System Models

December 2003
Notice

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

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

Warranty

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

Restricted Rights Legend

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

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


Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation.

Pentium® is a U.S. registered trademark of Intel Corporation.

PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of the Open Group.

Java™ is a U.S. trademark of Sun Microsystems, Inc.
# Contents

1 Amplifiers and Mixers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1-1</td>
</tr>
<tr>
<td>Curve-Fitting Algorithm</td>
<td>1-1</td>
</tr>
<tr>
<td>Amplifier (Obsolete RF System Amplifier)</td>
<td>1-3</td>
</tr>
<tr>
<td>Amplifier2 (RF System Amplifier)</td>
<td>1-11</td>
</tr>
<tr>
<td>AmplifierVC (Ideal Voltage-Controlled Amplifier)</td>
<td>1-35</td>
</tr>
<tr>
<td>AmpSingleCarrier (Single Carrier Amplifier)</td>
<td>1-36</td>
</tr>
<tr>
<td>FreqMult (Ideal Frequency Multiplier)</td>
<td>1-37</td>
</tr>
<tr>
<td>LogACDemod (Demodulating AC Logarithmic Amplifier)</td>
<td>1-39</td>
</tr>
<tr>
<td>LogDC (DC Logarithmic Amplifier)</td>
<td>1-40</td>
</tr>
<tr>
<td>LogSuccDetect (Successive Detection Logarithmic Amplifier)</td>
<td>1-41</td>
</tr>
<tr>
<td>LogTrue (True Logarithmic Amplifier)</td>
<td>1-42</td>
</tr>
<tr>
<td>Mixer (First RF System Mixer, Polynomial Model for Nonlinearity)</td>
<td>1-43</td>
</tr>
<tr>
<td>Mixer2 (Second RF System Mixer, Polynomial Model for Nonlinearity)</td>
<td>1-48</td>
</tr>
<tr>
<td>OpAmp (Operational Amplifier)</td>
<td>1-53</td>
</tr>
<tr>
<td>OpAmpIdeal (Ideal Operational Amplifier)</td>
<td>1-56</td>
</tr>
<tr>
<td>VMult (Voltage Multiplier)</td>
<td>1-58</td>
</tr>
</tbody>
</table>

2 Filters

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2-1</td>
</tr>
<tr>
<td>Filter Categories</td>
<td>2-1</td>
</tr>
<tr>
<td>References</td>
<td>2-5</td>
</tr>
<tr>
<td>BPF_Bessel (Bandpass Filter, Bessel-Thompson)</td>
<td>2-6</td>
</tr>
<tr>
<td>BPF_Butterworth (Bandpass Filter, Butterworth)</td>
<td>2-7</td>
</tr>
<tr>
<td>BPF_Chebyshev (Bandpass Filter, Chebyshev)</td>
<td>2-8</td>
</tr>
<tr>
<td>BPF_Elliptic (Bandpass Filter, Elliptic)</td>
<td>2-10</td>
</tr>
<tr>
<td>BPF_Gaussian (Bandpass Filter, Gaussian)</td>
<td>2-11</td>
</tr>
<tr>
<td>BPF_PoleZero (Bandpass Filter, Pole Zero)</td>
<td>2-12</td>
</tr>
<tr>
<td>BPF_Polynomial (Bandpass Filter, Polynomial)</td>
<td>2-14</td>
</tr>
<tr>
<td>BPF_RaisedCos (Bandpass Filter, Raised-Cosine)</td>
<td>2-15</td>
</tr>
<tr>
<td>BSF_Bessel (Bandstop Filter, Bessel-Thompson)</td>
<td>2-18</td>
</tr>
<tr>
<td>BSF_Butterworth (Bandstop Filter, Butterworth)</td>
<td>2-19</td>
</tr>
<tr>
<td>BSF_Chebyshev (Bandstop Filter, Chebyshev)</td>
<td>2-20</td>
</tr>
<tr>
<td>BSF_Elliptic (Bandstop Filter, Elliptic)</td>
<td>2-21</td>
</tr>
<tr>
<td>BSF_Gaussian (Bandstop Filter, Gaussian)</td>
<td>2-22</td>
</tr>
<tr>
<td>BSF_PoleZero (Bandstop Filter, Pole Zero)</td>
<td>2-23</td>
</tr>
<tr>
<td>BSF_Polynomial (Bandstop Filter, Polynomial)</td>
<td>2-24</td>
</tr>
<tr>
<td>BSF_RaisedCos (Bandstop Filter, Raised-Cosine)</td>
<td>2-25</td>
</tr>
<tr>
<td>HPF_Bessel (Highpass Filter, Bessel-Thompson)</td>
<td>2-28</td>
</tr>
<tr>
<td>HPF_Butterworth (Highpass Filter, Butterworth)</td>
<td>2-29</td>
</tr>
</tbody>
</table>
3 Modulators and Demodulators

Introduction ................................................................. 3-1
AM_DemodTuned (AM Demodulator, Tuned) .......... 3-3
AM_ModTuned (AM Modulator, Tuned) .................. 3-4
FM_DemodTuned (FM Demodulator, Tuned) .......... 3-5
FM_ModTuned (FM Modulator, Tuned) .................. 3-6
IQ_DemodTuned (I/Q Demodulator, Tuned) .......... 3-8
IQ_ModTuned (I/Q Modulator, Tuned) .................. 3-9
N_StateDemod (N-State Demodulator) ................. 3-10
N_StateMod (N-State Modulator) ......................... 3-11
PI4DQPSK_ModTuned (PI/4 DQPSK Modulator, Tuned) 3-13
PM_DemodTuned (PM Demodulator, Tuned) .......... 3-15
PM_ModTuned (PM Modulator, Tuned) ................. 3-16
PM_UnwrapDemodTuned (PM Unwrapped Demodulator, Tuned) 3-17
QPSK_ModTuned (QPSK Modulator, Tuned) .......... 3-18

4 Passive System Components

AntLoad (Antenna Load) ................................................. 4-2
Attenuator (Attenuator) ................................................ 4-4
Balun3Port (Balun, 3-port) .......................................... 4-9
Balun4Port (Balun, 4-port) .......................................... 4-11
Balun6Port (Balun, 6-port) .......................................... 4-13
Circulator (Ideal 3-Port Circulator) ....................... 4-14
CouplerDual (Dual Coupler) ....................................... 4-16
CouplerSingle (Single Coupler) ............................. 4-18
Gyrator (Gyrator) ...................................................... 4-20
Hybrid90 (Ideal 90-degree Hybrid Coupler) .............................................................. 4-21
Hybrid180 (Ideal 180-degree Hybrid Coupler) .......................................................... 4-23
IsolatorSML (SMLIsolator) ........................................................................................ 4-25
LOS_Link (Line-Of-Sight Antenna Link) ................................................................... 4-27
Pad (Pi or Tee Format) ............................................................................................. 4-29
PhaseShiftSML (Phase Shifter) ................................................................................ 4-31
PwrSplit2 (2-Way Power Splitter) .............................................................................. 4-33
PwrSplit3 (3-Way Power Splitter) .............................................................................. 4-34
TimeDelay (Time Delay) ........................................................................................... 4-35
Transformer (Ideal 4-Port Transformer) .................................................................... 4-36
TransformerG (Transformer with Ground Reference) .............................................. 4-37
TwoPort (2-Port Model) ............................................................................................. 4-38

5 Phase Lock Loop Components
DivideByN (Divide by N) ........................................................................................... 5-2
PhaseFreqDet (Frequency Detector, Baseband) ...................................................... 5-5
PhaseFreqDet2 (Frequency Detector, Baseband) ..................................................... 5-9
PhaseFreqDetCP (Frequency Detector, Baseband with Charge Pump) ................. 5-13
PhaseFreqDetTuned (Phase Frequency Detector, Tuned) ......................................... 5-14
PhaseNoiseMod (Phase Noise Modulator) ................................................................. 5-17
VCO (Voltage Controlled Oscillator) ......................................................................... 5-19
VCO_DivideByN (VCO Divide By N) ......................................................................... 5-22

6 Switch and Algorithmic Components
Comparator (Comparator) .......................................................................................... 6-2
ClockLFSR (Linear Feedback Shift Register) ............................................................. 6-3
Differentiator (Differentiator) .................................................................................... 6-7
DPDT_Static (Double Pole Double Throw Switch, Static) ......................................... 6-8
IntegratorSML (Integrator) ......................................................................................... 6-10
LimiterSML (Limiter) ................................................................................................. 6-11
ParallelSerial (Parallel to Serial Shift Register) ...................................................... 6-13
QuantizerSML (Quantizer) ......................................................................................... 6-14
ResetSwitch (Reset Switch) ....................................................................................... 6-16
SampleHoldSML (Sample Hold) ................................................................................ 6-17
Sampler (Sampler) .................................................................................................... 6-18
SerialParallel (Serial to Parallel Shift Register) ...................................................... 6-22
SPDT_Dynamic (Single Pole Double Throw Switch, Dynamic) .............................. 6-23
SPDT_Static (Single Pole Double Throw Switch, Static) .......................................... 6-25
SwitchV (Voltage Controlled Switch) ......................................................................... 6-27
SwitchV_Model (Voltage Controlled Switch Model) ................................................. 6-29
VSum (Voltage Summer) .......................................................................................... 6-30

7 Tx/Rx Subsystems
<table>
<thead>
<tr>
<th>Model Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF_PA_CKT (RF Power Amplifier Circuit)</td>
<td>7-2</td>
</tr>
<tr>
<td>RF_RX_SML (RF Receiver)</td>
<td>7-3</td>
</tr>
<tr>
<td>RF_TX_SML (RF Transmitter)</td>
<td>7-4</td>
</tr>
</tbody>
</table>

8 System Data Models

Introduction | 8-1
Classification of ADS System Data Models | 8-2
AmpH1H2 (Amplifier/Fundamental and 2nd Harmonic vs. Input Power) | 8-4
AmpH1H2_Setup (Amplifier/Fundamental and 2nd Harmonic vs. Input Power Setup) | 8-6
AmplifierP2D (P2D File Amplifier; FDD-Based, for Single Carrier Signal) | 8-8
AmplifierP2D_Setup (P2D File Amplifier; FDD-Based, for Single Carrier Signal Setup) | 8-11
AmplifierS2D (S2D File Amplifier, Polynominal Model for Nonlinearity) | 8-13
AmpLoadPull (SDD Load-Pull Amplifier) | 8-17
Balun3Port (Balun, 3-port) | 8-18
Balun4Port (Balun, 4-port) | 8-20
IQ_Demod_Data (IQ Demodulator Behavioral Model) | 8-22
IQ_Demod_Setup (IQ Demodulator Setup) | 8-24
IQ_Mod_Data (IQ Modulator Behavioral Model) | 8-26
IQ_Mod_Setup (IQ Modulator Setup) | 8-28
LoadPullSetup (Load Pull Setup) | 8-30
MixerHBdata (2-Tone HB Mixer) | 8-32
MixerHBsetup (2-Tone HB Mixer Setup) | 8-35
MixerIMT (Obsolete Intermodulation Table Mixer) | 8-37
MixerIMT2 (Intermodulation Table Mixer) | 8-42
VCA_Data (Voltage Controlled Amplifier) | 8-48
VCA_Setup (Voltage Controlled Amplifier Setup) | 8-51
Chapter 1: Amplifiers and Mixers

Introduction

The Filters - <filter type> and System - <device type> palettes contain two fundamentally different types of behavioral system models.

Filters, System - Amps & Mixers, and System - Mod/ Demod can be classified as tops-down system models that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as parameter-based behavioral models.

System - Data Models can be classified as bottoms-up system models that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as data-based behavioral models.

The parameter-based behavioral models typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The data-based behavioral models typically provide superior accuracy relative to the parameter-based behavioral models as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between parameter- and data-based behavioral models justify a palette emphasis on flow (all data-based behavioral model's grouped together) rather than functionality (all amplifiers, mixers, modulators, and demodulators grouped together) and resulted in the addition of a System - Data Models palette.

The use model for parameter-based behavioral models is to simply set a series of parameters prior to using the model. The use model for data-based behavioral models is slightly more involved. For a discussion, refer to Chapter 8, System Data Models.

Curve-Fitting Algorithm

The curve-fitting algorithm to determine the nonlinear behavior of the system mixer models is based on fitting a polynomial to the specified data where the saturation power (Psat) is calculated when the derivative of this polynomial is zero.
Amplifiers and Mixers

\[ P_n(x) = a_1 x_1 + a_2 x_2^2 + a_3 x^3 + \ldots \]

It is important to note that the coefficients \( a_4, a_6, a_8, \ldots \) are always zero. In only one case \( a_2 \) is non-zero and that's when SOI and TOI are specified.

Here are the different scenarios and what order of polynomial used to curve-fit.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOI</td>
<td>3</td>
</tr>
<tr>
<td>TOI &amp; AM2PM</td>
<td>3</td>
</tr>
<tr>
<td>SOI &amp; TOI</td>
<td>3, ( a_2 \neq 0 )</td>
</tr>
<tr>
<td>PndB</td>
<td>3</td>
</tr>
<tr>
<td>PndB &amp; AM2PM</td>
<td>3</td>
</tr>
<tr>
<td>Psat</td>
<td>5</td>
</tr>
<tr>
<td>TOI &amp; PndB</td>
<td>5</td>
</tr>
<tr>
<td>PndB &amp; Psat</td>
<td>7</td>
</tr>
<tr>
<td>TOI &amp; Psat</td>
<td>7</td>
</tr>
<tr>
<td>PndB &amp; TOI &amp; Psat</td>
<td>9</td>
</tr>
</tbody>
</table>
Amplifier (Obsolete RF System Amplifier)

Symbol

Note 2003C introduces an improved version of Amplifier: Amplifier2. Please use Amplifier2 for new designs; refer to Amplifier2 documentation for more information.

Parameters

S21 = forward transmission gain (real or complex number; refer to note 2)
S11 = port 1 reflection (real or complex number; refer to note 2)
S22 = port 2 reflection (real or complex number; refer to note 2)
S12 = reverse transmission gain (real or complex number; refer to note 2)
NF = input noise figure, in dB
NFmin = minimum noise figure at Sopt, in dB
Sopt = optimum source reflection for NFmin
Rn = equivalent noise resistance
Z1 = reference impedance for port 1 (real or complex number)
Z2 = reference impedance for port 2 (real or complex number)
ClipDataFile = clip data beyond maximum input power: NO (default) to disable; YES to enable
ImpNoncausalLength = non-causal function impulse response order
ImpMode = convolution mode
ImpMaxFreq = maximum frequency to which device is evaluated
ImpDeltaFreq = sample spacing in frequency
ImpMaxOrder = maximum allowed impulse response order
ImpWindow = smoothing window
ImpRelTol = relative impulse response truncation factor
Amplifiers and Mixers

ImpAbsTol = absolute impulse response truncation factor

Range of Usage
NF \geq 0 \text{ dB}
NF_{\text{min}} > 0
0 < |S_{\text{opt}}| < 1
0 < R_n
GainCompFreq > 0
For S_{21} = \text{mag/\text{ang}}
|S_{21}| > 0

Gain Compression Parameters
GainCompType = gain compression type:
LIST, use model gain compression specifications
FILE, use file-based gain compression data
GainCompFreq = reference frequency for gain compression (if gain compression is
described as a function of frequency)
ReferToInput = specify gain compression with respect to input or output power of
device
SOI = second order intercept, in dBm
TOI = third order intercept, in dBm
Psat = power level at saturation, in dBm
GainCompSat = gain compression at Psat, in dB
GainCompPower = power level in dBm at gain compression for X dB compression
point, specified by GainComp, in dBm
GainComp = gain compression at GainCompPower, in dB (default is 1 dB)
AM2PM = amplitude modulation to phase modulation, in degree/dB
PAM2PM = power level at AM2PM, in dBm
GainCompFile = filename for gain compression data in S2D file format
Range of Usage for Gain Compression Parameters

When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. The recommended parameter combinations are listed here.

**Note** If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

- Third-order intercept and 1dB gain compression parameters:
  TOI, GainCompPower with GainComp=1dB
  Range of validity: TOI > GainCompPower + 10.8

- Third-order intercept and power saturation parameters:
  TOI, Psat, GainCompSat
  Range of validity: TOI > Psat + 8.6

- 1dB gain compression and power saturation parameters:
  GainCompPower with GainComp=1dB, Psat, GainCompSat
  Range of validity: Psat > GainCompPower + 3

- Third-order intercept, 1dB gain compression and power saturation parameters:
  TOI, GainCompPower with GainComp=1dB, Psat, GainCompSat
  Range of validity: Psat > GainCompPower + 3, TOI > GainCompPower + 10.8

- Second-order intercept and third-order intercept parameters: SOI, TOI

- AM to PM with 1dB gain compression parameters:
  AM2PM, PAM2PM, and GainCompPower with GainComp=1dB
  The value for AM2PM must satisfy this condition to avoid a square root of a negative number:
  \[ \text{AM2PM} < \frac{180}{\pi} \times 10^{\left(\text{PAM2PM} - \text{GainCompPower}/10\right) \times 10^{-1} \times (\text{GainComp}/20)} \times \frac{1}{2.34} \]

- AM to PM with third-order intercept parameters:
  AM2PM, PAM2PM, and TOI
  The value for AM2PM must satisfy this condition to avoid a square root of a negative number:
Amplifiers and Mixers

If SOI is not specified, the amplifier is modeled using a polynomial of odd orders:

\[ y = a_1 x + a_3 x^3 + a_5 x^5 + \ldots \]

As a result, only odd order harmonics \((m \times f, \text{ where } m \text{ is an odd number})\) and odd order intermods \((m \times f_1 + n \times f_2, \text{ where } m+n \text{ is an odd number})\) are taken into account.

If SOI is specified, the amplifier polynomial has an even order term:

\[ y = a_1 x + a_2 x^2 + a_3 x^3 + a_5 x^5 + \ldots \]

As a result, both odd and even order harmonics and intermods are taken into account in the simulation.

**Warning Messages**

When values for TOI, 1 GainCompPower, and Psat are properly related, the DC input-output transfer characteristic has the form shown in Figure 1-1.

- **No Saturation.** A warning is displayed if a polynomial is generated that does not have a maximum where the transfer characteristics can be clipped (when the amplifier cannot reach saturation). Refer to Figure 1-2.

- **Non-Monotonic Transfer Curve.** A warning is displayed if the value specified for Psat is lower than GainCompPower. Refer to Figure 1-3.

The result of this specification is that the saturated output is lower than the output at the 1 dB compression point, and the input-output characteristics have a non-monotonic characteristic transfer curve.
Notes/Equations

1. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

\[
\frac{R_n}{Z_0} \geq \frac{T_0(F_{\text{min}} - 1)|1 + \text{Sopt}|^2}{T^4} \left(1 - |S_{11}|^2\right)^2
\]

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

2. Use the function polar(mag,ang), or dbpolar(dB, ang), or VSWRpolar(VSWR, ang) to convert these specifications into a complex number.

3. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).

   example: S21=dbpolar(10+ripple(),0.)

4. When you define the gain using S21, keep in mind that this gain is applied to the forward incident wave into the input of the amplifier. This is in keeping with the measurement standards used to define amplifier gain at a system level. This means that if you change S11 from 0 to 0.9 for example, you will see no change in output power because the reflect wave is not taken into account by the amplifier’s definition of gain, only the incident wave.

5. This model blocks dc.

6. For circuit envelope simulation, baseband signals are blocked.

7. OmniSys used GComp1-GComp7 data items for specifying gain compression. Table 1-1 summarizes the gain compression data for OmniSys and ADS. Refer to Figure 1-4 for OmniSys parameter information.

   GComp1-GComp6 can be specified by using the corresponding ADS gain compression parameters and setting GainCompType=LIST. Or, they can also be contained in an S2D format setting GainCompType=FILE. Also note that an S2D file could contain other data such as small signal S-parameters and noise; these data are ignored by the RF System Amplifier.
Table 1-1. Gain Compression Data for OmniSys and ADS

<table>
<thead>
<tr>
<th>OmniSys</th>
<th>ADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GComp1: IP3</td>
<td>TOI</td>
</tr>
<tr>
<td>GComp2: 1dBc</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td>GComp3: IP3, 1dBc</td>
<td>TOI</td>
</tr>
<tr>
<td>GComp4: IP3, Ps, GCS</td>
<td>TOI</td>
</tr>
<tr>
<td>GComp5: 1dBc, Ps, GCS</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td>GComp6: IP3, 1dBc, Ps, GCS</td>
<td>TOI</td>
</tr>
<tr>
<td>GComp7</td>
<td>GainCompType=FILE</td>
</tr>
</tbody>
</table>

8. The AM to PM option uses parabolic amplitude dependence to describe the amplitude to phase modulation conversion. When a signal of type \( V_{\text{in}}(\tau) = A\cos(\omega \tau) \), is applied to the input of a device with parabolic AM to PM, the output phase exhibits:
Therefore, this phase depends on the input signal amplitude $A$ in a parabolic manner. Because the conversion of amplitude to phase is amplitude dependent, the AM to PM (AM2PM) is specified in degrees per decibel at a given output power (PAM2PM). $k$ is calculated from these two parameters.

$$k = \frac{AM\,2\,PM \left( \frac{\pi}{180} \right)^2}{10} \cdot \frac{2.34}{(PAM\,2\,PM - 30)}$$

When AM to PM is specified, the third-order intermod and gain compression are side effects. If AM2PM is specified to be very large compared to the third-order intercept or gain compression, a warning is issued.
Amplifier2 (RF System Amplifier)

Symbol

Parameters

S21 = forward transmission gain (real or complex number)
S11 = forward reflection coefficient (real or complex number)
S22 = reverse reflection coefficient (real or complex number)
S12 = reverse transmission gain (real or complex number)
NF = input noise figure, in dB (NF mode used for NFmin=0)
NFmin = minimum noise figure at Sopt, in dB ((NFmin, Sopt, Rn) mode used for NFmin>0)
Sopt = optimum source reflection for NFmin ((NFmin, Sopt, Rn) mode used for NFmin>0)
Rn = equivalent noise resistance ((NFmin, Sopt, Rn) mode used for NFmin>0)
Z1 = reference impedance for port 1 (must be a real number)
Z2 = reference impedance for port 2 (must be a real number)
GainCompType = gain compression type:
  LIST: use model gain compression specifications (default)
  FILE: use file-based gain compression data
GainCompFreq = reference frequency for gain compression if gain compression is described as a function of frequency
ReferToInput = specify gain compression with respect to input or output power of device:
  OUTPUT: refer to output (default)
  INPUT: refer to input
SOI = second order intercept, in dBm
TOI = third order intercept, in dBm
Amplifiers and Mixers

Psat = power level at saturation, in dBm (always referred to output, regardless of the value of the ReferToInput parameter)
GainCompSat = gain compression at Psat, in dB (default in 5 dB)
GainCompPower = power level at gain compression for X dB compression point, specified by GainComp, in dBm
GainComp = gain compression at GainCompPower, in dB (default is 1 dB)
AM2PM = amplitude modulation to phase modulation, in degree/dB
PAM2PM = power level at AM2PM, in dBm
GainCompFile = filename for gain compression data in S2D file format
ClipDataFile = clip data beyond maximum input power:
  YES: enable (default)
  NO: disable
ImpNoncausalLength = non-causal function impulse response order
ImpMode = convolution mode
ImpMaxFreq = maximum frequency to which device is evaluated
ImpDeltaFreq = sample spacing in frequency
ImpMaxOrder = maximum allowed impulse response order
ImpWindow = smoothing window
ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Frequently Asked Questions
Q1: What are the major differences between Amplifier and Amplifier2?
A1: Refer to note 1.
Q2: What are the supported parameter combinations?
A2: Refer to "Range of Usage" on page 1-15.
Q3: What is the range of usage for each parameter combination?
A3: Refer to "Range of Usage" on page 1-15.
Q4: What model is being used?
A4: A polynomial model. Refer to "Modeling Basics" on page 1-16.
Q5: What polynomial order is being used?
A5: It depends on the parameters set on the component. Refer to “Modeling Basics” on page 1-16.

Q6: Can these polynomials blow up?
A6: No, these polynomials are limited. Refer to “Modeling Basics” on page 1-16 for details.

Q7: The parameters specified for Amplifier2 match those for my transistor level amplifier. Why don't the harmonics generated by Amplifier2 match those for my transistor level amplifier?
A7: Amplifier2 only matches the specified parameters. Refer to “Modeling Basics” on page 1-16 for details.

Q8: I have a small-signal and a large-signal tone. Why does the small-signal tone have a smaller gain than the large-signal tone?
A8: It is a consequence of the polynomial model adopted by Amplifier2. Refer to “Modeling Basics” on page 1-16 for details.

Q9: In a swept-power harmonic balance simulation, my higher-order harmonics are zero up to a certain order but then they suddenly experience a very sharp increase. Isn't this wrong?
A9: No, it's called generating a square wave via hard limiting. Refer to “Modeling Basics” on page 1-16 for details.

Q10: How do I validate AM to PM conversion?
A10: Refer to “AM to PM Conversion” on page 1-21.

Q11: Why does AM to PM conversion only match my transistor level amplifier around the power level PAM2PM at which I specified AM2PM?
A11: With information at only one power level, Amplifier2 has to rely on assumptions and generic curve shapes to model AM to PM at all power levels. For better modeling, use an S2D file and specify the exact magnitude and phase variation of your amplifier. Refer to “AM to PM Conversion” on page 1-21 for details.

Q12: With AM to PM conversion enabled, my output phase is supposed to be constant at high power levels. Why does my AM to PM start changing at high power levels?
A12: This is probably due to harmonic balance aliasing errors and can be mitigated by oversampling. Refer to “AM to PM Conversion” on page 1-21 for details.

Q13: With AM to PM conversion enabled, why don't the predicted second- and third-order intercept points match SOI and TOI as set on Amplifier2?
A13: It is a consequence of fitting the magnitude and phase separately. Refer to "AM to PM Conversion" on page 1-21 for details.

Q14: Why do Amplifier2's noise results differ from those for Amplifier?
A14: Amplifier2 models noise differently from Amplifier. Refer to "Noise" on page 1-24 for details.

Q15: What noise model is used by Amplifier2?
A15: Refer to "Noise" on page 1-24.

Q16: What is "NF-only mode" and "(NF min, Sopt, Rn) mode" for Amplifier2?
A16: It is two different ways of specifying noise. Refer to "Noise" on page 1-24 for details.

Q17: How is NF ssb/NF dsb calculated, both at low powers and in compression?
A17: Refer to "Noise" on page 1-24.

Q18: Can you provide more details about the noise voltages and noise figures produced by Amplifier2 for the different modes of operation?
A18: Refer to "Noise" on page 1-24.

Q19: With complex S21, why does Amplifier2 differ from Amplifier?
A19: Amplifier2 handles complex S21 values differently from Amplifier, leading to more physical waveforms. Refer to note 5 for details.

Q20: Why doesn't Amplifier2 work for complex reference impedances?
A20: Amplifier2 does not support complex reference impedances, per note 6.

Q21: Why is Amplifier2 sometimes slower than Amplifier?
A21: This is a consequence of the implementational differences between Amplifier and Amplifier2. Refer to note 9 for details.

Q22: Why don't the predicted second- and third-order intercept points match SOI and TOI as set on Amplifier2?
A22: You are probably not setting up your validation correctly. Make sure to do a two-tone simulation, not a one-tone simulation. Refer to note 10 for details.

Q23: Why does my ACPR level get better as I get deeper into compression?
A23: This is a consequence of the polynomial model adopted by Amplifier2. Refer to note 11 for details.

Q24: Why does Amplifier2 ignore the ACDATA block of my S2D file?
A24: It uses the S-parameters on the component instead. Refer to note 12 for details.
Q25: When the power range in an S2D file differs from that of a simulation, which power range is used for polynomial fitting?
A25: The power range in the S2D file. Refer to note 14 for details.

Q26: Why do I get ringing at high powers when using an S2D file?
A26: This is a consequence of the polynomial model adopted by Amplifier2. Refer to note 15 for details.

Q27: How do I get rid of this ringing?
A27: Eliminate data for high input powers, rely on extrapolation via ClipDataFile=yes, or break the S2D file into two. Worst case, use AmpSingleCarrier. Refer to note 15 for details.

Q28: Why do I see different results when I use the Analog/RF Amplifier2 component and the Ptolemy GainRF component with the same parameters?
A28: While based on the same OmniSys legacy, the implementations differ and thus the components can give different results. Refer to note 17 for details.

Range of Usage

\[
\begin{align*}
|S_{ij}| &> 0 \quad (i=1,2; j=1,2) \\
NF &\geq 0 \text{ dB} \\
NF_{\text{min}} &\geq 0 \text{ dB} \\
0 &< |S_{\text{opt}}| < 1 \\
R_n &> 0 \\
\text{GainCompFreq} &> 0
\end{align*}
\]

When specifying gain compression using model parameters, only certain combinations of parameters will produce stable polynomial curve fitting. If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations. The recommended parameter combinations are listed here:

- Third-order intercept parameter:
  Parameters: TOI
  Range of validity: N/A

- Gain compression parameters:
  Parameters: GainCompPower, GainComp
  Range of validity: N/A

- Power saturation parameters:
  Parameters: Psat, GainCompSat
  Range of validity: N/A
• Third-order intercept and 1dB gain compression parameters:
  Parameters: TOI, GainCompPower with GainComp=1dB
  Range of validity: TOI > GainCompPower + 10.8

• Third-order intercept and power saturation parameters:
  Parameters: TOI, Psat, GainCompSat
  Range of validity: TOI > Psat + 8.6

• 1dB gain compression and power saturation parameters:
  Parameters: GainCompPower with GainComp=1dB, Psat, GainCompSat
  Range of validity: Psat > GainCompPower + 3

• Third-order intercept, 1dB gain compression and power saturation parameters:
  Parameters: TOI, GainCompPower with GainComp=1dB, Psat, GainCompSat
  Range of validity: Psat > GainCompPower + 3, TOI > GainCompPower + 10.8

• Second-order intercept and third-order intercept parameters:
  Parameters: SOI, TOI
  Range of validity: N/A

**Modeling Basics**

Amplifier2 is based on a polynomial model of the magnitude of the output voltage as a function of the input voltage. If SOI is not specified, the magnitude response is modeled using a polynomial of odd orders

\[ y = a_1 \times x + a_3 \times x^3 + a_5 \times x^5 + \ldots \]

The order of the polynomial depends on the data entered by the user. If SOI is specified, the magnitude response has an even order term

\[ y = a_1 \times x + a_2 \times x^2 + a_3 \times x^3 \]

The order of the polynomial is hardwired at 3. The polynomial orders for the various magnitude modes are summarized in Table 1-2.

<table>
<thead>
<tr>
<th>Magnitude Mode</th>
<th>Polynomial Order</th>
<th>Polynomial Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOI</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SOI, TOI</td>
<td>3 (with second-order term)</td>
<td></td>
</tr>
<tr>
<td>GainCompPower, GainComp</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Psat, GainCompSat</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>TOI, GainComp=1dB, GainCompPower</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

1-16 Amplifier2 (RF System Amplifier)
For GainCompType=LIST, the phase response is described in the section “AM to PM Conversion” on page 1-21. For GainCompType=FILE and a GCOMP1-GCOMP6 block in the S2D file, the phase response is zero. For GainCompType=FILE and a GCOMP7 block in the S2D file, the phase response is modeled using an odd order polynomial, just like the magnitude response.

To prevent these polynomial expressions from blowing up and resulting in a non-physical model, the polynomial model is only used up to the largest \( x \) fulfilling \( \frac{dy}{dx} = 0 \), denoted by \( x_0 \). Above this value, the amplifier is hard-limited at \( y(x_0) \).

The polynomial coefficients for Amplifier2 are based on the parameters set for Amplifier2 regardless of the context in which Amplifier2 is used and the actual number of tones at the input of Amplifier2. These polynomial coefficients are then applied when Amplifier2 is analyzed in the environment where it lives. For example, given GainComp and GainCompPower, the polynomial coefficients are determined by exciting Amplifier2 with a one-tone signal and requiring the compressed output power to be GainComp below the linear response at GainCompPower. This can be confirmed by doing a one-tone analysis of Amplifier2. These polynomial coefficients are then used in any analysis using Amplifier2, whether it be a one-tone or a multi-tone analysis.

Amplifier2 matches the parameters that have been set on the component but is not guaranteed to match any other characteristics of a transistor level amplifier. Assume three parameters have been specified for Amplifier2: the linear gain \( S21 \), the 1 dB compression point \( \text{GainCompPower} \) (GainComp is set to 1 dB), and the third-order intercept TOI. This means that the behavior of Amplifier2 must match these parameters and nothing else. This is achieved by modeling \( y \) as a function of \( x \) via \( y = a_1 x + a_3 x^3 + a_5 x^5 \). The three known quantities allow the determination of the three unknown polynomial coefficients. If we look at this equation, we see that there is a fifth-order term. Therefore, Amplifier2 will produce fifth-order harmonics. However, this fifth-order order term is not supposed to match that from the transistor level amplifier from which the three parameters \( S21 \), \( \text{GainCompPower} \) and \( \text{TOI} \) were

<table>
<thead>
<tr>
<th>Magnitude Mode</th>
<th>Polynomial Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOI, Psat, GainCompSat</td>
<td>7</td>
</tr>
<tr>
<td>GainComp=1dB, GainCompPower, Psat, GainCompSat</td>
<td>7</td>
</tr>
<tr>
<td>TOI, GainComp=1dB, GainCompPower, Psat, GainCompSat</td>
<td>9</td>
</tr>
</tbody>
</table>
extracted. No information about the fifth-order intercept of that amplifier has been specified so we cannot match it. The fifth-order term for Amplifier2 is necessary to properly model $S_{21}$, GainCompPower and TOI and these parameters are properly modeled. In general, a transistor level amplifier will produce finite higher order harmonics which Amplifier2 does not produce.

For GainCompType=FILE and a GCOMP7 block in the S2D file, the magnitude and phase data in the GCOMP7 block of the S2D file is fitted to two separate polynomials. With $n$ power points in the GCOMP7 block, the polynomial order will be $\min(2 \times n - 1, 9)$. In most practical cases, the GCOMP7 block will have data at five or more power points and consequently ninth-order polynomials with odd-only terms will be used.

Consider now an amplifier with the response

$$y = a_1 x + a_2 x^2 + a_3 x^3$$

This is the model used when $S_{21}$, SOI and TOI are specified. If $a_2$ and $a_3$ are non-zero, they will be negative. When $a_2=0$, this is the model used when $S_{21}$ and TOI are specified. When $a_2=a_3=0$, this represents a linear amplifier. The behavior of this amplifier can be analyzed analytically when the excitation is simple enough. Doing so gives a good understanding of the behavior of more complicated amplifiers with different excitations.

Assuming the excitation

$$x(t) = A_1 \cos(\omega_1 t)$$

we get the response

$$y(t) = \frac{1}{2} a_2 A_1^2$$

$$+ \left( a_1 A_1 + \frac{3}{4} a_3 A_1^3 \right) \cos(\omega_1 t)$$

$$+ \frac{1}{2} a_2 A_1^2 \cos(2 \omega_1 t)$$

$$+ \frac{1}{4} a_3 A_1^3 \cos(3 \omega_1 t)$$

and can define the gain of the fundamental $A_1 \cos(\omega_1 t)$ as

$$G_1 = a_1 + \frac{3}{4} a_3 A_1^2$$
Assuming the excitation
\[ x(t) = A_1 \times \cos(\omega_1 \times t) + A_2 \times \cos(\omega_2 \times t) \]
we get the response
\[ y(t) = \frac{1}{2} \times a_2 \times (A_1^2 + A_2^2) \]
\[ + \left( a_1 \times A_1 + \frac{3}{4} \times a_3 \times A_1^3 + \frac{3}{2} \times a_3 \times A_1 \times A_2^2 \right) \times \cos(\omega_1 \times t) \]
\[ + \left( a_1 \times A_2 + \frac{3}{4} \times a_3 \times A_2^3 + \frac{3}{2} \times a_3 \times A_2 \times A_1^2 \right) \times \cos(\omega_2 \times t) \]
\[ + \frac{1}{2} \times a_2 \times A_1^2 \times \cos(2 \times \omega_1 \times t) \]
\[ + \frac{1}{2} \times a_2 \times A_2^2 \times \cos(2 \times \omega_2 \times t) \]
\[ + \frac{1}{4} \times a_3 \times A_1^3 \times \cos(3 \times \omega_1 \times t) \]
\[ + \frac{1}{4} \times a_3 \times A_2^3 \times \cos(3 \times \omega_2 \times t) \]
\[ + a_2 \times A_1 \times A_2 \times \cos((\omega_1 - \omega_2) \times t) \]
\[ + a_2 \times A_1 \times A_2 \times \cos((\omega_1 + \omega_2) \times t) \]
\[ + \frac{3}{4} \times a_3 \times A_2 \times A_1^2 \times \cos((2 \times \omega_1 + \omega_2) \times t) \]
\[ + \frac{3}{4} \times a_3 \times A_2 \times A_1^2 \times \cos((2 \times \omega_1 - \omega_2) \times t) \]
\[ + \frac{3}{4} \times a_3 \times A_1 \times A_2^2 \times \cos((2 \times \omega_2 + \omega_1) \times t) \]
\[ + \frac{3}{4} \times a_3 \times A_1 \times A_2^2 \times \cos((2 \times \omega_2 - \omega_1) \times t) \]
and can define the gain of the fundamentals \( A_1 \times \cos(\omega_1 \times t) \) and \( A_2 \times \cos(\omega_2 \times t) \) as
\[ G_1 = a_1 + a_3 \times \left( \frac{3}{4} \times A_1^2 + \frac{3}{2} \times A_2^2 \right) \]
and
Amplifiers and Mixers

\[ G_2 = a_1 + a_3 \times \left( \frac{3}{2} \times A_1^2 + \frac{3}{4} \times A_2^2 \right) \]

respectively.

Examining \( G_1 \) and \( G_2 \) closer, we see that \( G_1 > G_2 \) for \( A_1 > A_2 \) and \( a_3 < 0 \). This means that if we excite a polynomial amplifier with a large- and a small-signal tone, the large-signal tone will experience a larger gain than the small-signal tone. For more details, please refer to “Polynomial Model of Blocker Effects on LNA/Mixer Devices” by W. Domino et. al. in the June 2001 issue of Applied Microwave and Wireless.

This behavior is counter-intuitive for some people. If we excite an amplifier with a small-signal tone, it will provide its maximum/linear gain. If we excite the same amplifier with a large-signal tone, it will provide a smaller/compressed gain. If we excite an amplifier with both a large- and a small-signal tone, one could then expect that the small-signal tone would be subject to a larger gain than the large-signal tone. The above shows that this is not how Amplifier2 behaves.

This behavior, however, is intuitive for some people. If we consider an ideal hard limiting amplifier with an input voltage of \( V_{\text{in}} \) and an output voltage of \( V_{\text{max}} \), it has a large-signal gain of \( V_{\text{max}}/V_{\text{in}} \) but a small-signal gain of zero, suggesting a larger large-signal gain than small-signal gain. This is in line with how Amplifier2 behaves.

Above, we presented the polynomial model used for modeling the output voltage as a function of the input voltage. This polynomial model, however, does not fully describe the response of the amplifier. It is important to remember that the amplifier has a power level at which it saturates and above which the output is clipped. We also presented the result of pushing \( A_1 \times \cos(\omega_1 \times t) \) and \( A_1 \times \cos(\omega_1 \times t) + A_2 \times \cos(\omega_2 \times t) \) signals through a \( y = a_1 \times x + a_2 \times x^2 + a_3 \times x^3 \) nonlinearity. The result is an output voltage consisting of a finite number of \( C_n \times \cos(n \times \omega_n \times t) \) terms. More generally, it can easily be shown that pushing a sum of \( A_i \times \cos(\omega_i \times t) \) terms through a polynomial nonlinearity will result in a finite number of \( C_n \times \cos(n \times \omega_n \times t) \) terms. It might then be expected that a harmonic balance analysis of such an amplifier can only produce finite harmonics up to a certain order but will always produce zero harmonics beyond a certain order. This is the case when the amplifier is not hard limiting but is not the case when the amplifier starts hard limiting.

When we analyze an amplifier via a harmonic balance simulation of a certain order, the harmonics we will see on the input and output of the amplifier will be those which lead to the most correct input and output waveforms that can be achieved with the given order.
Assume the input waveform is a sine wave. This can be represented exactly using only the fundamental. Therefore, we will see a finite fundamental tone and zero harmonics on the input.

The output waveform depends on the input power level.

At low input powers, the output waveform is simply a sine wave, namely the input sine wave scaled by the linear amplifier gain. This can be represented exactly using only the fundamental. Therefore, we will see a finite fundamental tone and zero harmonics.

At higher input powers where the amplifier enters compression, the output waveform is the input sine wave pushed through the amplifier polynomial. Because a sine wave pushed through a polynomial gives rise to sine terms and no others, the output waveform will have finite harmonics up to a certain order and zero harmonics above that order.

At high enough input powers that the amplifier starts hard limiting, the output waveform is a clipped sine wave. For increasingly high input powers, this clipped sine wave approaches a square wave. Representation of such a clipped wave requires finite third, fifth, seventh, etc. harmonics so when the amplifier starts clipping all harmonics will be finite. This means that we will see a very sharp increase in the level of all such harmonics as soon as the amplifier starts clipping. If we increase the input power enough, we get approximately a square wave output. When this happens, the levels of the various harmonics reach saturation. The relative values of these saturated power levels can be predicted from the Fourier representation of a square wave.

**AM to PM Conversion**

The Amplifier component supports AM to PM conversion for a very limited number of magnitude modes; Amplifier2 supports AM to PM conversion for all magnitude modes when GainCompType=LIST. Magnitude and phase is fitted separately so AM to PM conversion can be added for each magnitude mode. This adds a phase response but does not alter the magnitude response.

AM to PM conversion is defined as the amount of phase change in degrees per magnitude/power change in dB. It is specified via the two parameters AM2PM and PAM2PM. AM2PM [degrees/dB] is the amount of AM to PM conversion while PAM2PM [dBm] is the power level at which the amplifier has this amount of AM to PM conversion. A physically sensible phase response is then constructed which has a phase of phase(S21) at low powers, an amount of AM to PM conversion given by AM2PM at PAM2PM, constant phase at high powers and for which the maximum
Amplifiers and Mixers

The amount of AM to PM at any power level is also given by AM2PM. Stated differently, the derivative of the phase response with respect to power in dB takes its maximum value AM2PM at PAM2PM.

This phase response is far from unique and certainly will not be correct for all amplifiers. Based only on the amount of AM to PM at one power level, there is simply no way to construct a phase response that matches any circuit level amplifier at all power levels. This AM to PM capability is only meant as a crude way of incorporating a phase response for amplifiers about which very little information is known - at the initial system level design iterations. For example, where no transistor level amplifier has been designed. Once more information about the amplifier's phase response is known and a more accurate phase modeling is desired, use Amplifier2 with GainCompType=FILE and specify the (compression and) phase response as a function of power in the GCOMP7 block of an S2D file. This is much more accurate than what this AM to PM capability can provide.

To document the exact phase response in a little more detail, we start out with AM2PM [degrees/dB] and PAM2PM [dBm]. We first define

\[ k = \text{AM2PM} \]
\[ P = 10^{(\text{PAM2PM} - 30)/10} \]

For a sine wave
\[ A_0 \cos(\omega t) \]
the average power is

\[ P = \frac{1}{\text{period}} \int (A_0 \cos(\omega t))^2 dt = \frac{A_0^2}{2} \]

Therefore, we have

\[ A_0 = \sqrt{2 \times P} \]

The phase response is initially modeled in the A-domain. We choose

\[ y(A) = c_0 + c_1 \times A + c_2 \times A^2 + c_3 \times A^3 \]

We now get

\[ y(0) = 0 \Rightarrow c_0 = 0 \]
\[ dy/dA(0) = 0 \Rightarrow c_1 = 0 \]
\[ dy/dA(A_0) = k \Rightarrow 2 \times A_0 \times c_2 + 3 \times A_0^2 \times c_3 = 0 \]
$$\frac{d^2y}{dA^2}(A_0) = 0 \rightarrow c_2 + 3 \times A_0 \times c_3 = 0$$

\[ c_2 = k/A_0 \]

\[ c_3 = k/(3 \times A_0^2) \]

Then, we have

\[ y(A) = k/A_0 \times A^2 - k/(3 \times A_0^2) \times A^3 \]

This function has the following properties

\[ y(0) = 0 \]

\[ \frac{dy}{dA}(0) = 0 \]

\[ y(A_0) = 2/3 \times k \times A_0 \]

\[ \frac{dy}{dA}(A_0) = k \]

\[ \frac{d^2y}{dA^2}(A_0) = 0 \text{ (} A_0 \text{ is an extremum, max AM to PM occurs here)} \]

\[ y(2 \times A_0) = 4/3 \times k \times A_0 = 2 \times y(A_0) \]

\[ \frac{dy}{dA}(2 \times A_0) = 0 \]

This curve will be distorted when it is made a function of power in dBm instead of magnitude A. The difference between two log values can be quite significant at small values but will be much smaller at large values. This means that the log function will distort the curve - it will be stretched out at low powers and will be compressed at high powers. This will change the point where the maximum slope occurs so the curve is shifted/scaled such that the maximum slope AM2PM occurs at PAM2PM. Also, we account for the fact that the derivative with respect to A must be in dB and not linear numbers.

The above phase response applies to the fundamental tone. This phase shift is applied before the application of the nonlinear polynomial which means that the change of the phase response of the nth order harmonic from its low power value will equal that for the fundamental tone scaled by the harmonic index n.

The easiest way to validate that the amount of AM to PM conversion takes its maximum value AM2PM at PAM2PM is to use the `diff` function in ADS. If the Amplifier2 output voltage is denoted by Vout and determined via a swept input power harmonic balance simulation, simply plot `diff(phase(Vout[:,1]))`. The peak of this curve should occur at PAM2PM and should take the value AM2PM.

All harmonic balance simulations can be subject to aliasing errors. This can significantly change the phase response of an amplifier. To mitigate the effects of harmonic balance aliasing errors, increase the Fundamental Oversample (FundOversample) parameter on the HarmonicBalance controller from the default 1 to, say, 16.
The magnitude and phase response for Amplifier2 are fitted separately. Given, say, SOI and TOI, a set of magnitude polynomial coefficients are determined such that Amplifier2 has a second-order intercept point of SOI and a third-order intercept point of TOI. Given AM2PM and PAM2PM, a set of phase polynomial coefficients are determined such that Amplifier2 has AM to PM conversion given by AM2PM at PAM2PM. When using an Amplifier2 with both a magnitude and a phase response in a multi-tone harmonic balance simulation, there is no guarantee that the above magnitude and phase separation will hold. Generally, a phase response will generate intermodulation distortion with a multi-tone input. This is part of why AM to PM conversion is of interest. This means that the third-order intercept point may no longer be the TOI set on the component. Amplifier2 makes no attempt at matching the composite magnitude/phase third-order intercept point to the TOI set on the component but simply uses the polynomial coefficients derived for the isolated magnitude mode.

**Noise**

Given the minimum noise figure NFmin (real), the optimal reflection coefficient Sopt (complex), the noise resistance Rn (real), the noise reference impedance Rref (real), and the source admittance Ys (complex), the noise figure NF of an amplifier is determined by

\[
Y_{\text{opt}} = \frac{1}{R_{\text{ref}}} \times \frac{1 - \text{Sopt}}{1 + \text{Sopt}}
\]

\[
F_{\text{min}} = 10^{\frac{\text{NFmin}}{10}}
\]

\[
F = F_{\text{min}} + \frac{\text{Rn}}{\text{Real}[Y_s]} \times |Y_s - Y_{\text{opt}}|^2
\]

\[
\text{NF} = 10 \times \log(F)
\]

Note that this is independent of the amplifier S-parameters.

The noise behavior of Amplifier2 is characterized by the four noise parameters NF, NFmin, Sopt and Rn and the reference impedance Z1 for port 1. Amplifier2 is implemented as a Noisy2Port cascaded with an SDD. The above-mentioned five parameters control the parameters for the Noisy2Port, the Noisy2Port generates a noise voltage on its output and this noise voltage is passed through the SDD in the same manner as the signal.

NF-only mode is used for NFmin=0. This is a special case where only one noise parameter must be specified. In this case, the Noisy2Port has the parameters NFmin=NF, Sopt=0, and Rn=Z1/4 \times (10^{NF/10.1}). The reference impedance for the
Amplifier2 (RF System Amplifier) 1-25

noise calculation (not available on the Noisy2Port user interface) is \(Z_1\). The \(\text{NF}_{\min} = 0\), \(\text{Sopt}\) and \(\text{Rn}\) parameters are ignored.

\((\text{NF}_{\min}, \text{Sopt}, \text{Rn})\) mode is used for \(\text{NF}_{\min} > 0\). This is a more general case than \(\text{NF-Only mode}\). In this case, the Noisy2Port has the parameters \(\text{NF}_{\min} = \text{NF}_{\min}\), \(\text{Sopt} = \text{Sopt}\), and \(\text{Rn} = \text{Rn}\). The reference impedance for the noise calculation (not available on the Noisy2Port user interface) is \(Z_1\). The NF parameter is ignored.

Given an output noise voltage \(v_n\), the single sideband noise figure \(\text{NF}_{\text{ssb}}\) and the double sideband noise figure \(\text{NF}_{\text{dsb}}\) are given by

\[
\text{NF}_{\text{ssb}} = \frac{v_n^2/R + k \times T_0 \times (G1 + G2 + ...)}{(k \times T_0 \times (G1 + G2 + ...))} / (k \times T_0 \times (G1 + G2 + ...))
\]

\[
\text{NF}_{\text{dsb}} = \frac{v_n^2/R + k \times T_0 \times (G1 + G2 + ...)}{(k \times T_0 \times (G1 + G2 + ...))} / (k \times T_0 \times (G1 + G2 + ...))
\]

where \(R\) is the output resistance, \(k = 1.380658 \times 10^{-23} \text{ J/K}\) is Boltzmann's constant, \(T_0 = 290 \text{ K}\) is the standard noise temperature, \(G1\) is the primary power gain from the input noise frequency to the output noise frequency, and \(G2+...\) is the sum of all higher order mixing gains which mix from some input frequency to the output noise frequency. For an amplifier, \(G2+...\) is zero under small-signal operation. \(v_n^2/R\) represents the noise added by the amplifier while \(k \times T_0\) represents the noise power available from the input termination. In the following, we outline how Amplifier2 calculates noise voltages and noise figures in various cases. This will be compared to the behavior of Amplifier.

**Small-Signal Operation, NF-Only Mode**

Consider an amplifier with \(\text{NF} = 5 \text{ dB}\), \(Z1 = 25 \text{ Ohm}\) and \(S21 = 1\). The corresponding values of \(\text{NF}_{\min}\), \(\text{Sopt}\) and \(\text{Rn}\) have been described previously. We will consider the two \(S11\) values \(S11 = 0\) and \(S11 = 0.2\). The amplifier lives in an environment where \(Ys = 1/50 \text{ Siemens}\) and \(R = 50 \text{ Ohm}\). For this amplifier, the noise figure on the output should be \(\text{NF} = 5.356 \text{ dB}\).

Amplifier calculates the noise voltage via

\[
\text{an}^2 = k \times T_0 \times \left(\text{NF}_{\min} - 1\right)
\]

\[
\text{bn}^2 = S21^2 \times \text{an}^2
\]

\[
v_n = \sqrt{R \times \text{bn}^2}
\]

Doing the numerical evaluations, we get \(v_n = 657.93 \text{ pV}\). This value is independent of \(S11\). Because the \(G1\) term used in the calculation of \(\text{NF}_{\text{ssb}}\) and \(\text{NF}_{\text{dsb}}\) varies with
Amplifiers and Mixers

S11, NFssb and NFdsb will vary with S11. Specifically, we get NFssb=NFdsb=5.356 dB for S11=0 (matches NF) and NFssb=NFdsb=4.940 dB for S11=0.2 (does not match NF).

Amplifier2 calculates the noise voltage differently. First, four noise correlation coefficients are calculated via

\[ \langle v_n, v_n \rangle = 4 \times k \times T_0 \times R_n \]
\[ \langle i_n, i_n \rangle = 4 \times k \times T_0 \times R_n \times | Y_{opt} |^2 \]
\[ \langle v_n, i_n \rangle = 4 \times k \times T_0 \times R_n \times ( ( F_{min} - 1 ) / ( 2 \times R_n ) ) \times Y_{opt} \]
\[ \langle i_n, v_n \rangle = \text{conj}(\langle v_n, i_n \rangle) \]

The noise voltage can then be calculated from

\[ v_n = \sqrt{G \times \langle v_n, v_n \rangle \times G + G \times \langle v_n, i_n \rangle \times Z + Z \times \langle i_n, v_n \rangle \times G + Z \times \langle i_n, i_n \rangle \times Z} \]

where G is the trans-voltage-gain of the amplifier and Z is the trans-impedance of the amplifier. These can be found from an adjoint circuit analysis of the amplifier and become G=0.4714 and Z=23.5702 Ohm for S11=0 and G=0.5051 and Z=25.2538 Ohm for S11=0.2. The noise voltage then becomes vn=657.93 pV for S11=0 and vn=704.93 pV for S11=0.2. We see that vn varies with S11; this is not surprising. The noise voltage at the output of the Noisy2Port is independent of S11 but this voltage is passed through the SDD and will see a small-signal gain. NFssb and NFdsb become 5.356 dB independent of S11 (matches NF).

Small-Signal Operation, (NFmin, Sopt, Rn) Mode

Consider an amplifier with NFmin=5 dB, Sopt=0.1+j × 0.2, Rn=40, Z1=25 Ohm and S21=1. We will consider the two S11 values S11=0 and S11=0.2. The amplifier lives in an environment where Ys=1/(10-j × 20) Siemens and Zl=30. For this amplifier, the noise figure on the output should be NF = 9.520 dB.

In this case, Amplifier2 calculates its noise voltage from the same equations as for NF-only mode. The only difference is the NFmin, Sopt and Rn values used. Amplifier2 produces the noise voltage vn=766.79 pV for S11=0 and vn=749.68 pV for S11=0.2. For both S11 values, this leads to NFssb=NFdsb=9.520 dB. Amplifier calculates noise voltages different from NF-only mode but is again wrong. Amplifier produces the noise voltage vn=611.47 pV for S11=0 and vn=597.83 pV for S11=0.2. For both S11 values, this leads to NFssb=NFdsb=7.823 dB.

Large-Signal Operation

The above sections present the noise behavior of Amplifier and Amplifier2 in NF-only mode and (NFmin, Sopt, Rn) mode. These apply as the amplifier is operated under
small-signal conditions. As the power is increased and the amplifier is operated under large-signal conditions, the expressions for NFssb and NFdsb still apply but some of the terms change.

Consider the expressions for NFssb and NFdsb. These are

\[
\begin{align*}
\text{NFssb} &= \left( \frac{v_n^2}{R} + k \times T_0 \times (G_1 + G_2 +...) \right) / (k \times T_0 \times G_1) \\
\text{NFdsb} &= \left( \frac{v_n^2}{R} + k \times T_0 \times (G_1 + G_2 +...) \right) / (k \times T_0 \times (G_1 + G_2 +...))
\end{align*}
\]

At low powers, \(G_1\) is the amplifier's small-signal gain and \(G_2+...\) is zero. As the power increases and the amplifier compresses, \(G_1\) decreases (gain expansion could precede this trend) and \(G_2+...\) increases. This is a function of the signal properties of the amplifier and is independent of the amplifier's noise model. The variation of \(G_1\) and \(G_2+...\) as a function of the power level is the same for Amplifier and Amplifier2. This is not the case for the noise voltage \(v_n\). For Amplifier, \(v_n\) is constant as the amplifier compresses. For Amplifier2, the noise voltage at the output of the Noisy2Port will pass through the SDD which means that \(v_n\) decreases as the amplifier compresses. This lowers \(v_n^2/R\) for Amplifier2 relative to Amplifier, causing Amplifier2 to produce smaller noise figures in compression than Amplifier.

Notes/Equations

1. Amplifier2 is introduced as a replacement for Amplifier. To change an existing Amplifier component to an Amplifier2 component, simply change the name from Amplifier to Amplifier2 on the schematic. The parameters for the two models are the same and Amplifier2 will adopt the values for Amplifier, making parameter re-entry unnecessary. The only exception is that the parameters GainCompType, ReferToInput and ClipDataFile will take their default values LIST, OUTPUT and yes, respectively, regardless of the values these parameters had for Amplifier. Also, the Amplifier display settings will be ignored. Amplifier2 will simply adopt its default settings, displaying S21, S11, S22 and S12.

For examples of how to use Amplifier2, replace Amplifier2 in ADS examples where Amplifier is currently used. To locate such examples, use the ADS documentation Search feature, Search in: Examples, and enter Amplifier. As each found item is highlighted, the ADS examples project directory information will be displayed; navigate through the ADS Main window > File > Example Project to these examples.

Table 1-3 summarizes the major differences between Amplifier and Amplifier2.
Amplifiers and Mixers

Table 1-3. Major Differences between Amplifier and Amplifier2

<table>
<thead>
<tr>
<th>Amplifier</th>
<th>Amplifier2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM to PM only supported for some magnitude modes</td>
<td>AM to PM supported for all magnitude modes</td>
</tr>
<tr>
<td>AM to PM broken</td>
<td>AM to PM working</td>
</tr>
<tr>
<td>Non-physical noise behavior</td>
<td>Physical noise behavior</td>
</tr>
<tr>
<td>Real/imaginary polynomial fit</td>
<td>Magnitude/phase polynomial fit</td>
</tr>
<tr>
<td>Complex S21 leads to non-physical time-domain waveforms</td>
<td>Complex S21 leads to physical time-domain waveforms</td>
</tr>
</tbody>
</table>

For large harmonic balance and circuit envelope simulations Amplifier2 may be slower than Amplifier.

2. Use the functions polar(mag, ang), dbpolar(dB, ang), or VSWRpolar(VSWR, ang) to convert the Sij specifications into complex numbers. Note that Sij are voltage gains and not power gains. For example, an amplifier with $S_{21} = \text{polar}(10,0)$ and $S_{11}=S_{22}=S_{12}=0$ will scale the voltage by a factor of 10 from input to output and will therefore result in a 20 dB increase in power. $S_{21} = \text{dbpolar}(10,0)$, on the other hand, will result in a 10 dB increase in power.

3. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple (mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).

   example: $S_{21} = \text{dbpolar}(10.0+\text{ripple}(),0.0)$

4. When defining gain using $S_{21}$, remember that this gain is applied to the forward incident wave into the input of the amplifier. This is in keeping with the measurement standards used to define amplifier gain at the system level. This means that if we change $S_{11}$ from 0 to 0.9 for example, we will see no change in output power because the reflected wave is not taken into account by the amplifier’s definition of gain, only the incident wave.

5. Amplifier2 behaves differently from Amplifier for complex $S_{21}$ values. The phase shift is applied before the nonlinear polynomial instead of after, leading to much more realistic waveforms. To see this, consider for example an Amplifier and Amplifier2 with $S_{21} = \text{dbpolar}(10,50)$, TOI =20 dBm, GainCompPower=10 dBm and GainComp=2 dB. Excite and terminate these with default P_1Tone and Term components and carry out a high-order harmonic balance analysis, say Order=150, at a high input power, say 30 dBm. Because the amplifiers are excited with a sine wave and operate deep into
compression, we expect the output time domain waveform to be clipped and
closely resemble a square wave. Except for the expected differences at
transitions where the effect of a finite Order is evident, this is the case for
Amplifier2 but is not at all the case for Amplifier.

6. Amplifier2 does not support complex reference impedances.

7. Amplifier2 blocks dc.

8. For circuit envelope simulation, baseband signals are blocked.

9. Amplifier2 may be slower than Amplifier for large harmonic balance and circuit
envelope simulations.

10. “Modeling Basics” on page 1-16 presented the polynomial model used for
modeling the output voltage as a function of the input voltage. It also presented
the result of pushing an \( A_1 \times \cos(\omega_1 \times t) + A_2 \times \cos(\omega_2 \times t) \) signal through a
\( y = a_1 \times x + a_2 \times x^2 + a_3 \times x^3 \) nonlinearity. Based on this, we can calculate the IP2
(second-order intercept) and IP3 (third-order intercept) of the amplifier. The
general equation for the nth-order intercept point IPn is
\[ IP_n = \frac{n \times P_1 - P_n}{n-1} \]
where \( P_1 \) is the power level of the first-order tone and \( P_n \) is the power level of
the nth-order tone.

The \( P_n \) power level, however, is not unique. Look at the amplifier output.
Ignoring \( A_1 \) and \( A_2 \) which we can normalize out, the second-order harmonics
\( \cos(2 \times \omega_1 \times t) \) and \( \cos(2 \times \omega_2 \times t) \) have the amplitude \( 1/2 \times a_2 \) while the
second-order intermods \( \cos((\omega_1+\omega_2) \times t) \) and \( \cos((\omega_1-\omega_2) \times t) \) have the
amplitude \( a_2 \). Similarly, the third-order harmonics \( \cos(3 \times \omega_1 \times t) \) and
\( \cos(3 \times \omega_2 \times t) \) have the amplitude \( 1/4 \times a_3 \) while the third-order intermods
\( \cos((2 \times \omega_1+\omega_2) \times t) \), \( \cos((2 \times \omega_1-\omega_2) \times t) \), \( \cos((2 \times \omega_2+\omega_1) \times t) \) and
\( \cos((2 \times \omega_2-\omega_1) \times t) \) have the amplitude \( 3/4 \times a_3 \). If the formula IPn =
\( (n \times P_1-P_n)/(n-1) \) works for one type of second- and third-order tone, it will not
work for the other.

The industry standard definitions of IP2 and IP3 are based on intermods, not
harmonics. When polynomial coefficients are determined in the Amplifier2 code,
they are therefore based on intermods. When validating Amplifier2 using
intermods, everything checks out; when using harmonics, it does not. This
shows that we cannot validate the proper SOI and TOI for Amplifier2 using a
one-tone analysis because this generates only harmonics but not intermods. We
use industry standard definitions when deriving the coefficients for Amplifier2
and it is imperative to do the same when validating Amplifier2.
Note that SOI and TOI are defined at a low power level. If SOI and TOI are calculated at a power level where either P1 or Pn deviate from their low-power values, the results will be in error. To see this, sweep the input power and plot the IP3 of the Amplifier2 as a function of the input power. At low powers, the IP3 of the amplifier will match the TOI parameter set on the Amplifier2. Amplifier2 was constructed to have the TOI parameter as its IP3 at low powers so this is expected. As the input power increases, IP3 will start to deviate from the TOI value for low input powers. This simply reinforces the importance of calculating IP3 at low input powers. Too high, and IP3 changes. Note that it is not enough that the fundamental tone varies linearly. SOI/TOI is calculated based on the fundamental and the second-/third-order intermods so one must ensure that the second-/third-order intermods are also linear or SOI/TOI will change.

Note also that Amplifier2 must be output impedance matched in order for the SOI/TOI validation to check out.

11. The common expectation for the behavior of the adjacent channel power rejection (ACPR) for an amplifier operating in compression can be formulated in many equivalent ways: the ACPR decreases/improves as TOI increases; the ACPR decreases/improves as the amplifier becomes more linear; the ACPR decreases/improves as we move away from compression; the ACPR increases/worsens as TOI decreases; the ACPR increases/worsens as the amplifier becomes more nonlinear; the ACPR increases/worsens as we move towards compression.

This expectation is correct for an amplifier operating linearly. For an amplifier operating in compression, the validity of this expectation depends on the amplifier. We will discuss two cases, a transistor-level amplifier and a polynomial amplifier.

The expectation may or may not be correct for a transistor-level amplifier operating in compression. A transistor-level simulation of a particular amplifier in ADS shows that this expectation holds at some input power ranges but not at others.

The expectation is incorrect for a polynomial amplifier operating in compression. "Modeling Basics" on page 1-16 presented how two tones react when pushed through a polynomial amplifier and offered a discussion of this analysis. The result is that the large-signal gain will always exceed the small-signal gain. Applied to ACPR skirts, this means that a polynomial amplifier will actually produce a decreasing/better ACPR level as we move
further into compression. This may be counter-intuitive but Amplifier2 reacts the way we can theoretically predict that a polynomial amplifier must act. Amplifier2’s behavior is simply a function of the polynomial modeling on which it is based.

12. An S2D file typically consists of an ACDATA block containing small-signal information and a GCOMPi block (i=1, ..., 7) containing compression information. For Amplifier2, the ACDATA block is ignored and the S-parameters specified on the Amplifier2 component are used. If the S-parameters of the ACDATA block must be used, use the AmplifierS2D component instead. Similarly, any NDATA blocks containing noise data are ignored by Amplifier2.

13. When an S2D file contains gain compression data at more than one frequency, the GainCompFreq must be set to one of the frequencies in the S2D file to identify the data to be used. It is imperative that GainCompFreq be set to one of the frequencies in the S2D file as no interpolation or extrapolation between gain compression data at different frequencies can be performed. For further details regarding GainCompFreq selectivity, refer to Table 8-2 in the AmplifierS2D documentation.

14. When an S2D file has a power range that exceeds that of a simulation, a choice must be made for the power range used for fitting. Assume an S2D file covers -30 dBm to 30 dBm but that a simulation is carried out from -10 dBm to 10 dBm. In this case, a choice must be made as to whether the polynomial fitting of S2D data is done over the power range -30 dBm to 30 dBm or -10 dBm to 10 dBm. In the former case, the fitting may be inaccurate as the polynomial must cover a large power range that could hold a lot of variations. This is undesirable. However, the advantage of this approach is that the results we get when simulating from -10 dBm to 10 dBm are a subset of what we would have gotten in that interval had we simulated from -30 dBm to 30 dBm. In the latter case, the fitting is much more accurate as the fitting is done over a much smaller power range which presumably holds a lot less variation. This is desirable. However, the problem with this approach is that the results between -10 dBm and 10 dBm will be different for a simulation done from -30 dBm and 30 dBm rather than from -10 dBm and 10 dBm because the polynomial coefficients change as we change the power range of the simulation. ADS does the former. It fits a polynomial to the whole S2D file, not just the subset for which the simulation is carried out. To change the fitting in a power range, it is not enough to change the power range of the simulation. To change the fitting, one must modify the S2D file. The S2D file power range, not the simulation.
Amplifiers and Mixers

power range, dictates the fitting power range. This is relevant in the following where we discuss different fittings in different power ranges.

15. A typical Pout (output power) vs. Pin (input power) curve consists of a linearly increasing region, a transition region and a saturation region. Another way of thinking of this is that typical Pout-Pin vs. Pin curve consists of a flat region, a transition region and a linearly decreasing region.

When the saturation region is made larger and larger, the fitting approach adopted by Amplifier2 (polynomial fitting, odd order terms, order dependent upon the number of data points in the S2D file, max order 9) will tend to produce fitted curves which ring/oscillate more and more at higher powers. Mild ringing is often tolerable and might not even be noticed but if the transition region becomes too large it can make the results useless. To alleviate the problem, reduce the size of the saturation region to the minimum needed and leave no extra points in the S2D file. If the results are still not satisfactory, ensure ClipDataFile is set to yes and reduce the saturation region even more, relying on Amplifier2 extrapolation. If the results are still not satisfactory, try breaking the S2D file into two files and simulate the problem in two steps. Another alternative is to use the AmpSingleCarrier model. This model is based on linear interpolation instead of curve fitting and will not have this ringing problem. AmpSingleCarrier, however, will not produce harmonics. Refer to the AmpSingleCarrier documentation for details.

If fitted results do not accurately match the data in the S2D file and it is uncertain if this ringing problem is the cause, the problem is very easy to exaggerate. Simply extend the GCOMP7 block of the S2D file with a large flat region (more input powers, saturated output power, saturated output phase) and re-simulate. If the ringing problem is the cause, the results should get worse.

16. The S2D file capability is a legacy from OmniSys. OmniSys used GComp1-GComp7 data items for specifying gain compression. Table 1-4 summarizes the gain compression data for OmniSys and ADS. Refer to Figure 1-5 for OmniSys parameter information. GComp1-GComp6 can be specified by using the corresponding ADS gain compression parameters and setting GainCompType=LIST or they can be contained in an S2D format setting GainCompType=FILE.
Table 1-4. Gain Compression Data for OmniSys and ADS

<table>
<thead>
<tr>
<th>OmniSys</th>
<th>ADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GComp1: IP3</td>
<td>TOI</td>
</tr>
<tr>
<td>GComp2: 1dBc</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td></td>
<td>GainCompPower</td>
</tr>
<tr>
<td>GComp3: IP3, 1dBc</td>
<td>TOI</td>
</tr>
<tr>
<td></td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td></td>
<td>GainCompPower</td>
</tr>
<tr>
<td>GComp4: IP3, Ps, GCS</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td></td>
<td>Psat</td>
</tr>
<tr>
<td></td>
<td>GainCompSat</td>
</tr>
<tr>
<td>GComp5: 1dBc, Ps, GCS</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td></td>
<td>Psat</td>
</tr>
<tr>
<td></td>
<td>GainCompSat</td>
</tr>
<tr>
<td>GComp6: IP3, 1dBc, Ps, GCS</td>
<td>GainComp=1dB</td>
</tr>
<tr>
<td></td>
<td>Psat</td>
</tr>
<tr>
<td></td>
<td>GainCompSat</td>
</tr>
<tr>
<td>GComp7</td>
<td>GainCompType=FILE</td>
</tr>
<tr>
<td></td>
<td>GainCompFile=filename</td>
</tr>
</tbody>
</table>

Figure 1-5. OmniSys Parameter Information
17. The GainRF component is the Ptolemy equivalent of the Analog/RF Amplifier2 component. Both of these components are based on the OmniSys legacy (refer to note 16) and the OmniSys cases GCOMP1 through GCOMP7 are shared by GainRF and Amplifier2. For GCOMP1 through GCOMP6 corresponding to GainCompType=LIST, the curve fitting algorithms for GainRF and Amplifier2 are very similar and close results can be expected. However, the curve fitting algorithms are not identical and the shape of the knees of the compression curves will therefore differ slightly. Also, the levels at which the various fundamentals saturate can be different. These levels will generally differ more when Psat is not set than when Psat is set. For GCOMP7 corresponding to GainCompType=FILE, the curve fitting algorithms for GainRF and Amplifier2 are different and different results can be expected.
AmplifierVC (Ideal Voltage-Controlled Amplifier)

Symbol

Parameters
Gain = voltage gain, in dB/V
Rout = output resistance, in ohms

Range of Usage
Rout > 0

Notes/Equations
1. AmplifierVC is an ideal voltage controlled amplifier. The impedance for its input and control port is infinite; its output impedance is Rout.

2. An equation is used to describe Gain as a function of the control voltage at port 3: $V_3$. The default equation is $\text{Gain} = 30 - 15 \times V_3$. 
Amplifiers and Mixers

**AmpSingleCarrier (Single Carrier Amplifier)**

**Symbol**

![Symbol](image)

**Parameters**

- **Freq** = frequency
- **S21** = forward transmission coefficient; use x+j*y polar (x,y), dbpolar (x, y) for complex value
- **S11** = forward reflection coefficient; use x+j*y polar (x,y), dbpolar (x, y), vswrpolar (x, y) for complex value
- **S22** = reverse reflection coefficient; use x+j*y polar (x,y), dbpolar (x, y), vswrpolar (x, y) for complex value
- **S12** = reverse reflection coefficient; use x+j*y polar (x,y), dbpolar (x, y) for complex value
- **NF** = noise figure, in dB
- **NFmin** = minimum noise figure at Sopt, in dB
- **Sopt** = optimum source reflection for minimum noise figure; use x+j*y polar (x,y), dbpolar (x, y) for complex value
- **Rn** = equivalent noise resistance
- **Z1** = reference impedance for port 1
- **Z2** = reference impedance for port 2
- **GainCompFile** = filename for Gain Compression (GCOMP7) Data

**Notes/Equations**

1. AmpSingleCarrier is based on FDD (frequency-domain defined device). The incident power level at the amplifier input is detected and the corresponding gain is obtained by interpolating the gain compression table given by GainCompFile. In harmonic balance simulation, the output signal has only one frequency component given by the Freq parameter. Neither harmonics nor intermodulation products are generated by AmpSingleCarrier. In Envelope simulation, the output signal contains only the complex envelope around one frequency given by the Freq parameter.
FreqMult (Ideal Frequency Multiplier)

Symbol

Parameters

- $S_{11} =$ complex reflection coefficient for port 1
- $S_{22} =$ complex reflection coefficient for port 2
- $G_1 =$ power gain of input tone, in dB
- $G_2 =$ power gain of second harmonic relative to input tone, in dB
- $G_3 =$ power gain of third harmonic relative to input tone, in dB
- $G_4 =$ power gain of fourth harmonic relative to input tone, in dB
- $G_5 =$ power gain of fifth harmonic relative to input tone, in dB
- $G_6 =$ power gain of sixth harmonic relative to input tone, in dB
- $G_7 =$ power gain of seventh harmonic relative to input tone, in dB
- $G_8 =$ power gain of eighth harmonic relative to input tone, in dB
- $G_9 =$ power gain of ninth harmonic relative to input tone, in dB
- $P_{\text{min}} =$ minimum input power for specified conversion, in dBm
- $Z_1 =$ reference impedance for port 1
- $Z_2 =$ reference impedance for port 2

Range of Usage

- $0 \leq |S_{11}| < 1$
- $0 \leq |S_{22}| < 1$

Notes/Equations

1. The ideal frequency multiplier takes an input signal and produces an output spectrum with specified spectral harmonics. The reverse isolation is assumed to be infinite ($S_{12}=0$). All of the harmonics generation is specified relative to the input level. For example if an input power of 20 dBm is incident on a multiplier with $G_2=-20$ dB the second harmonic output will be 0 dBm. This device is compatible with transient simulation.
2. This model assumes that only one signal tone is present at the input. If multiple tones are used at the input then unwanted mixing products can be generated and spurious mixing products will result.

3. The harmonic balance parameter ORDER must be set to a value equal to or higher than the harmonic index of interest.

4. Real-world nonlinear devices such as mixers and frequency multipliers often have an input power level below which they do not work. For FreqMult, this phenomenon is incorporated through the PMin parameter. However, PMin is not simply a minimum threshold value to which the input power is limited.

   With a right-propagating input wave of $a_1$, the input power detection is done via $|a_1|^2 = a_1 \times a_1 + H(a_1) \times H(a_1)$, with $H(a_1)$ being the Hilbert transform of $a_1$. A normalized $a_1$ is then calculated via $a_1\text{norm}=a_1/\sqrt{|a_1|^2+\text{dBm2lin}(\text{PMin})}$, with $\text{dBm2lin}(x)=10^{(x-30)/10}$. Note the presence of PMin. For $\text{dBm2lin}(\text{PMin})<<|a_1|^2$, the effect of PMin is negligible. For $|a_1|^2$ approaching $\text{dBm2lin}(\text{PMin})$, however, the results will depend on the value of PMin. If this is undesired, simply lower PMin appropriately.
**LogACDemod (Demodulating AC Logarithmic Amplifier)**

**Symbol**

![Symbol for LogACDemod](image)

**Parameters**
- CurrentSlope = gradient of transfer characteristic, in amperes/decade
- VoltIntercept = Vin for zero output, in volts
- Z1 = reference impedance for port 1, in ohms

**Notes/Equations**
1. LogACDemod uses a square-law detector and an ideal logarithmic function.
2. The function is in the form of voltage-input/current-output.
3. LogACDemod is not recommended for transient simulation.
LogDC (DC Logarithmic Amplifier)

Symbol

Parameters

- VoltSlope = gradient of transfer characteristic, in volts/decade
- VoltIntercept = Vin for zero output, in volts
- Z1 = reference impedance for port 1, in ohms

Notes/Equations

1. LogDC provides an output that is a logarithmic function of the input. If, for example, the scaling is 1 volt/decade, the output changes by 1V for any tenfold increase of the input.

   The intercept point is the input level at which the output voltage is 0.

   Vout = VoltSlope \times \log((Vin / VoltIntercept) + 1)
LogSuccDetect (Successive Detection Logarithmic Amplifier)

Symbol

![Symbol Diagram]

Parameters

- **NumStages**: number of stages
- **StageGain**: gain per stage, in dB
- **CurrentSlope**: gradient of transfer characteristic, in amperes/decade
- **Z1**: reference impedance for port 1, in ohms

Notes/Equations

1. This amplifier uses a successive detection scheme to provide an output current proportional to the logarithm of the input voltage. The amplifier consists of several amplifier/limiter stages (NumStages specifies how many), each having a small signal gain (StageGain specifies the gain). Each stage has an associated full-wave detector, whose output current depends on the absolute value of its input stage. These output currents are summed to provide the output scaled at the CurrentSlope (amp/decade). The output contains amplitude information only, regardless of any phase information. A simplified block diagram of this component is shown in Figure 1-6.

![Block Diagram]

Figure 1-6. Simplified Block Diagram of a 5-Stage LogSuccDetect
Amplifiers and Mixers

LogTrue (True Logarithmic Amplifier)

Symbol

Parameters

- NumStages = number of stages
- StageGain = gain per stage, in dB
- VoltLimit = limiting voltage of each stage, in volts
- \( Z1 = \) reference impedance for port 1, in ohms

Notes/Equations

1. LogTrue accepts inputs of either polarity and generates an output whose sign follows that of the input. A progressive compression technique is used, in which the logarithmic response can be approximated through the use of cascaded amplifier stages that have signal-dependent gain.
Mixer (First RF System Mixer, Polynomial Model for Nonlinearity)

Symbol

Parameters

SideBand = produce UPPER, LOWER, or BOTH sidebands
ImageRej = image rejection at output with respect to fundamental, in dB
LO_Rej1 = LO to input rejection for LO leakage, in dB
LO_Rej2 = LO to output rejection for LO leakage, in dB
RF_Rej = input to output rejection for direct RF feedthrough, in dB
ConvGain = conversion gain (real or complex number; see note 2)
S11 = port 1 reflection (real or complex number; see note 2)
S22 = port 2 reflection (real or complex number; see note 2)
S33 = port 3 reflection (real or complex number; see note 2)
PminLO = minimum LO power threshold, in dBm
NF = input double sideband noise figure, in dB
NFmin = minimum double sideband noise figure at Sopt, in dB
Sopt = optimum source reflection for NFmin
Rn = equivalent noise resistance, in ohms
Z1 = reference impedance for port 1 (real or complex number)
Z2 = reference impedance for port 2 (real or complex number)
Z3 = reference impedance for port 3 (real or complex number)
ImpNoncausal Length = non-causal function impulse response order
ImpMode = convolution mode
ImpMaxFreq = maximum frequency at which device is evaluated
ImpDeltaFreq = sample spacing in frequency
Amplifiers and Mixers

ImpMaxOrder = maximum allowed impulse response order
ImpWindow = smoothing window
ImpRelTol = relative impulse response truncation factor
ImpAbsTol = absolute impulse response truncation factor

Range of Usage
NF ≥ 0 dB
NFmin > 0
0 < |Sopt| < 1
0 < Rn
GainCompFreq > 0
|ConvGain| > 0

Gain Compression Parameters
GainCompType = gain compression type: LIST, use model gain compression specifications; FILE, use file-based gain compression data
GainCompFreq = reference frequency for gain compression (if gain compression is described as a function of frequency)
ReferToInput = specify gain compression with respect to input or output power of device
SOI = second order intercept, in dBm
TOI = third order intercept, in dBm
Psat = power level at saturation, in dBm
GainCompSat = gain compression at Psat, in dB
GainCompPower = power level in dBm at gain compression specified by GainComp, in dBm
GainComp = gain compression at GainCompPower, in dB (default is 1dB)
GainCompFile = filename for gain compression data in S2D file format

Range of Usage for Gain Compression Parameters
When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. Recommended parameter combinations are listed here.
**Note** If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

- Third-order intercept and 1dB gain compression parameters: TOI, GainCompPower, with GainComp=1dB. Range of validity: TOI > GainCompPower + 10.8.
- 1dB gain compression and power saturation parameters: GainCompPower, with GainComp=1dB, and Psat, GainCompSat. Range of validity: Psat > GainCompPower + 3.
- Third-order intercept, 1dB gain compression and power saturation parameters: TOI, GainCompPower with GainComp=1dB, and Psat, GainCompSat. Range of validity: Psat > GainCompPower + 3, TOI > GainCompPower + 10.8
- Second-order intercept and third-order intercept parameters: SOI, TOI.

**Notes/Equations**

1. The Mixer component is similar to Mixer2. The key difference is that Mixer supports frequency conversion AC analysis or FCAC analysis for small-signal AC or S-parameter analysis, while Mixer2 does not. This capability allows small-signal frequency traditionally done at only one frequency to be somewhat extended to deal with more than one frequency. In terms of convergence, Mixer2 is typically more robust than Mixer, as the power detection is implemented differently.

2. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

\[
\frac{R_n}{Z_0} \geq \frac{TOI(F\text{min} - 1)|1 + \text{Sopt}|^2}{T^4} \frac{(1 - |S_{11}|^2)}{|1 - \text{Sopt}S_{11}|^2}
\]

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.
3. Use the function polar(mag,ang) or dbpolar(db, ang), or VSWR polar(VSWR, ang) to convert these specifications into a complex number.

4. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple (mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).

   Example: S21=dbpolar(10+ripple(),0.)

5. This model passes DC in the sense that a DC source at the RF input passes through the mixer to give a signal at the IF output.

6. In harmonic balance simulations, the PminLO parameter sets a threshold for the effect of the LO power on the mixer’s conversion gain. The default value of PminLO is -100 dBm. The mixer will provide the expected conversion gain if the LO power is significantly greater (~20 dB) than the value of PminLO. If the LO power is less than this amount, the mixer’s conversion gain will deteriorate in a nonlinear fashion.

7. Gain compression can be specified by using the gain compression model parameters, or this information can be contained in an S2D format file. All S2D gain compression types are supported by this model. Gain compression types 1 through 6 can also be described using the gain compression model parameters. Gain Compression 7 information must be contained in an S2D file. The GainCompType parameter instructs the model where to look for this data—in an S2D file or use model parameters.

   For S2D data file format information refer to “Working with Data Files” in the Circuit Simulation manual.

8. Mixer operation with the SOI and TOI parameters is described in this note.

   First consider an amplifier. Consider two input tones at f1 and f2 (assume f1<f2) and at the same power level. Do a 2-tone HB simulation. The output will have first-order tones at f1 and f2. It also have second-order intermod products at f2-f1 and at f1+f2. It will also have third-order intermod products at 2×f1-f2 (will be smaller than f1 by f2-f1) and 2×f2-f1 (will be greater than f2 by f2-f1). Drawing a picture of this frequency plan can be helpful. Based on these tones we can calculate the SOI and TOI of the amplifier. The general equation is \( IP_n = (n \times P1-P_n)/(n-1) \) where P1 is the power level of the first-order tone and Pn is the power level of the nth-order tone. For SOI, we can use the power levels at f1 and f2-f1 or we can use the power levels at f2 and f1+f2. For TOI, we can use the power levels at f1 and 2×f1-f2 or we can use the power levels at f2 and 2×f2-f1.
For a mixer, everything is the same, except that everything is being shifted down (down-converting mixer) or up (up-converting mixer) by the LO frequency fLO due to mixing. Therefore, all these frequencies will either have fLO subtracted or added. Note that this means that from an HB point of view, we're calculating SOI off of second- and third-order intermod products and calculating TOI off of second- and fourth-order intermod products. For example, TOI can be calculated from a P1 at f1-fLO (second-order as far as the HB simulation is concerned) and a Pn at 2 \times f1-f2-fLO (fourth-order as far as the HB simulation is concerned).

Also, note that the mixer must be output-matched in order to validate SOI and TOI.

For more information about SOI and TOI, refer to the Amplifier2 component documentation.

9. Frequency Conversion AC (FCAC) analysis requires a single frequency at every node of a circuit. If Sideband=BOTH is specified for Mixer, the simulator will use SideBand=LOWER for FCAC analysis.

10. When Mixer is used in Transient analysis for single sideband applications, it is recommended that the user set Sideband=BOTH and insert a high-order filter to suppress the undesired sideband.
Amplifiers and Mixers

**Mixer2 (Second RF System Mixer, Polynomial Model for Nonlinearity)**

**Symbol**

![Mixer Symbol]

**Parameters**

- **SideBand** = produce UPPER, LOWER, or BOTH sidebands
- **ConvGain** = conversion gain (real or complex number; see note 2)
- **SP11** = RF port reflection; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP12** = IF port to RF port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP13** = LO port to RF port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP21** = RF port to IF port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **S22** = IF port reflection; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP23** = LO port to IF port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP31** = RF port to LO port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP32** = IF port to LO port leakage; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **SP33** = LO port reflection; use $x+j*y$, polar $(x,y)$, dbpolar $(x,y)$, vswrpolar $(x,y)$, for complex value
- **PminLO** = minimum LO power threshold, in dBm
- **NF** = input double sideband noise figure, in dB
- **NFmin** = minimum double sideband noise figure at Sopt, in dB
So\text{pt} = \text{optimum source reflection for NF}_{\text{min}} \\
R_{\text{n}} = \text{equivalent noise resistance, in ohms} \\
Z_{1} = \text{reference impedance for port 1 (real or complex number)} \\
Z_{2} = \text{reference impedance for port 2 (real or complex number)} \\
Z_{3} = \text{reference impedance for port 3 (real or complex number)} \\
\text{ImpNoncausal Length} = \text{non-causal function impulse response order} \\
\text{ImpMode} = \text{convolution mode} \\
\text{ImpMaxFreq} = \text{maximum frequency at which device is evaluated} \\
\text{ImpDeltaFreq} = \text{sample spacing in frequency} \\
\text{ImpMaxOrder} = \text{maximum allowed impulse response order} \\
\text{ImpWindow} = \text{smoothing window} \\
\text{ImpRelTol} = \text{relative impulse response truncation factor} \\
\text{ImpAbsTol} = \text{absolute impulse response truncation factor} \\

\textbf{Range of Usage} \\
NF \geq 0 \text{ dB} \\
NF_{\text{min}} > 0 \\
0 < | So\text{pt} | < 1 \\
0 < R_{n} \\
GainCompFreq > 0 \\
| \text{ConvGain} | > 0 \\

\textbf{Gain Compression Parameters} \\
GainCompType = \text{gain compression type: LIST, use model gain compression specifications; FILE, use file-based gain compression data} \\
ReferToInput = \text{specify gain compression with respect to input or output power of device} \\
SOI = \text{second order intercept, in dBm} \\
TOI = \text{third order intercept, in dBm} \\
Psat = \text{power level at saturation, in dBm} \\
GainCompSat = \text{gain compression at Psat, in dB}
Amplifiers and Mixers

GainCompPower = power level in dBm at gain compression specified by GainComp, in dBm
GainComp = gain compression at GainCompPower, in dB (default is 1dB)
AM2PM = amplitude to phase modulation in degrees, dB
PAM2PM = power level at AM2PM in degrees
GainCompFile = filename for gain compression data in S2D file format

Range of Usage for Gain Compression Parameters
When specifying gain compression using model parameters, only certain combination of parameters will produce stable polynomial curve fitting. The recommended parameter combinations are listed here.

Note: If unrealistic parameter values are used, the polynomial will become unstable, resulting in oscillations.

• Third-order intercept and 1dB gain compression parameters: TOI, GainCompPower, with GainComp=1dB.

• Third-order intercept and power saturation parameters: TOI, Psat, GainCompSat.
  Range of validity: TOI > Psat + 8.6.

• 1dB gain compression and power saturation parameters: GainCompPower, with GainComp=1dB, and Psat, GainCompSat.
  Range of validity: Psat > GainCompPower + 3.

• Third-order intercept, 1dB gain compression and power saturation parameters: TOI, GainCompPower with GainComp=1dB, and Psat, GainCompSat.

• Second-order intercept and third-order intercept parameters: SOI, TOI.

Notes/Equations
1. The Mixer component is very similar to Mixer2. The key difference is that Mixer supports frequency conversion AC analysis or FCAC analysis for small-signal AC or S-parameter analysis, while Mixer2 does not. This capability allows small-signal frequency traditionally done at only one frequency to be...
somewhat extended to deal with more than one frequency. In terms of convergence, Mixer2 is typically more robust than Mixer, as the power detection is implemented differently.

2. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

\[
\frac{R_n}{Z_o} \geq \frac{T_o (F_{min} - 1) |1 + Sopt|^2}{T^4} \left( \frac{(1 - |S_{11}|)^2}{|1 - Sopt \cdot S_{11}|^2} \right)
\]

A warning message will be issued if Rn does not meet this criterion. If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to zero and a warning message will be issued.

3. Use the function polar(mag,ang) or dbpolar(dB, ang), or VSWR polar(VSWR, ang) to convert these specifications into a complex number.

4. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple(mag, intercept, period, variable); for example ripple(0.1, 0, 10 MHz, freq).

Example: S21=dbpolar(10+ripple(),0.)

5. This model passes DC in the sense that a DC source at the RF input is filtered out and does not pass through the mixer at the IF output. There can be a DC IF output signal for this mixer, but this results from the mixing signal and not a DC signal at FR input.

6. In harmonic balance simulations, the PminLO parameter sets a threshold for the effect of the LO power on the mixer’s conversion gain. The default value of PminLO is -200 dBm. The value of PminLO, treated as the available source power, is first converted to the source voltage as \(\sqrt{8 \times \text{Re}(Z3) \times 10^{(P_{\text{minLO}}-30)/10}}\) which is then used as a reference value in the mixer detection law formulas. Due to different implementations of Mixer2 and Mixer, the Mixer2 default value of -200 roughly corresponds to setting PminLO=72 in Mixer (for Z3 = 50 Ohm). The mixer will provide the expected conversion gain if the LO power is significantly greater (~20 dB) than the approximate value of PminLO/2 + 30, that is more than -50 dBm for the default setting. If the LO power is less than this amount, the mixer’s conversion gain will deteriorate in a nonlinear fashion.
7. Gain compression can be specified by using the gain compression model parameters, or this information can be contained in an S2D format file. All S2D gain compression types are supported by this model. Gain compression types 1 through 6 can also be described using the gain compression model parameters. Gain Compression 7 information must be contained in an S2D file. The GainCompType parameter instructs the model where to look for this data—in an S2D file or use model parameters.

For S2D data file format information refer to “Working with Data Files" in the Circuit Simulation manual.

8. Mixer2 operation with the SOI and TOI parameters is described in this note.

First consider an amplifier. Consider two input tones at f1 and f2 (assume f1<f2) and at the same power level. Do a 2-tone HB simulation. The output will have first-order tones at f1 and f2. It also have second-order intermod products at f2-f1 and at f1+f2. It will also have third-order intermod products at 2 × f1-f2 (will be smaller than f1 by f2- f1) and 2 × f2-f1 (will be greater than f2 by f2-f1). Drawing a picture of this frequency plan can be helpful. Based on these tones we can calculate the SOI and TOI of the amplifier. The general equation is IPn= (n × P1-Pn)/(n-1) where P1 is the power level of the first-order tone and Pn is the power level of the nth-order tone. For SOI, we can use the power levels at f1 and f2-f1 or we can use the power levels at f2 and f1+f2. For TOI, we can use the power levels at f1 and 2 × f1-f2 or we can use the power levels at f2 and 2 × f2-f1.

For a mixer, everything is the same, except that everything is being shifted down (down-converting mixer) or up (up-converting mixer) by the LO frequency fLO due to mixing. Therefore, all these frequencies will either have fLO subtracted or added. Note that this means that from an HB point of view, we're calculating SOI off of second- and third-order intermod products and calculating TOI off of second- and fourth-order intermod products. For example, TOI can be calculated from a P1 at f1-fLO (second-order as far as the HB simulation is concerned) and a Pn at 2 × f1-f2-fLO (fourth-order as far as the HB simulation is concerned).

Also, note that the mixer must be output-matched in order to validate SOI and TOI.

For more information about SOI and TOI, refer to the Amplifier2 component documentation.
OpAmp (Operational Amplifier)

Symbol

Parameters
Gain = open loop dc gain of amplifier, in dB
CMR = common mode rejection ratio, in dB
Rout = output resistance, in ohms
RDiff = differential input resistance, in ohms
CDiff = differential input capacitance, in farads
RCom = common mode input resistance, in ohms
CCom = common mode input capacitance, in farads
SlewRate = signal slew rate, in volts/sec
IOS = input offset current, in amperes
VOS = input offset voltage, in volts
BW = gain bandwidth product (unity gain bandwidth), in hertz
Pole1 = dominant pole frequency, in hertz (overrides BW parameter)
Pole2 = additional higher order pole frequency, in hertz
Pole3 = additional higher order pole frequency, in hertz
Pole4 = additional higher order pole frequency, in hertz
Pole5 = additional higher order pole frequency, in hertz
Zero1 = feed forward zero frequency, in hertz
Inoise = input spectral noise current, in amperes/sqrt(hertz)
Vnoise = input spectral noise voltage, in volts/sqrt(hertz)
VEE = negative supply voltage, in volts
VCC = positive supply voltage, in volts
Amplifiers and Mixers

DeltaVCC = difference between output saturation and VCC (see note 4)
DeltaVEE = difference between output saturation and VEE (see note 4)

Range of Usage
RDiff > 0
CDiff > 0
RCom > 0
CCom > 0
by default:
CMR = \infty
Pole2 = \infty
Pole3 = \infty
Pole4 = \infty
Pole5 = \infty
Zero1 = \infty

Notes/Equations
1. The BW parameter is the GainBandwidth product, i.e. it is the frequency at which the gain is unity or 0 dB. Pole1 is the basic amplifier pole and corresponds to the frequency where the gain starts sloping downward.

BW and Pole1 can be specified simultaneously; however, if both are entered, Pole1 will override BW, and Pole1 must then be entered as BW/Gain. The Gain parameter is the open loop gain of the opamp and it must be converted out of dB only for use in setting Pole1, i.e., Gain=10^{Gain\_dB/20}.

2. To match the phase shift from the data sheet, adjust the values of Pole2 through Pole5.

3. Zero1 is used for operational amplifiers with feed-forward or lead-lag compensation networks.

4. Output voltage is generally less than the rail voltage (VCC and VEE). Use DeltaVCC and DeltaVEE to specify the difference between the rail voltage and actual output voltage. For example, if VCC is +5V and the positive output is +4.5V, set DeltaVCC to 0.5V.

5. This opamp is a nonlinear model. If your circuit cannot achieve convergence using this model, use the OpAmpIdeal linear model.

6. The relationship between input and output voltages is given in the equation:

\[ Im \times \tanh(Vin/Im) = Vout/A0 - 10 \times (Vclip-Vout) + d/dt (Vout \times Tau1/A0) \]
where

- $A_0$ is open loop DC gain
- $V_{clip} = V_{out}$ as long as it is not limiting

$Im = \text{SlewRate} \times \frac{Tau1}{A_0}$

$\text{Tau1}=\frac{A_0}{2\pi/BW}$ when Pole1=0, otherwise $\text{Tau1}=\frac{1}{2\pi/BW}$
**OpAmplIdeal (Ideal Operational Amplifier)**

**Symbol**

![OpAmplIdeal Symbol](symbol.png)

**Parameters**

- **Gain**: magnitude of open loop dc voltage gain; use the `polar()` function to specify magnitude and phase
- **Z1**: input impedance, inverting terminal, in ohms
- **Z2**: input impedance, non-inverting terminal, in ohms
- **Z3**: output impedance, in ohms
- **Z4**: leakage impedance, inverting to non-inverting terminal, in ohms
- **Freq3db**: frequency at which gain magnitude is down by 3dB, in hertz
- **Delay**: time delay associated with gain, in seconds

**Range of Usage**

by default:

- **Z1** = $\infty$
- **Z2** = $\infty$
- **Z3** = 0
- **Z4** = $\infty$
- **Freq3db** = $\infty$

**Notes/Equations**

1. $V_s = (V^+ - V^-) Gain \times \frac{e^{j2\pi F Delay}}{F \frac{Freq 3dB}{1 + jF}}$

   $V_s = (V^+ - V^-) Gain (\text{for } f=0)$

   where $F$ is the simulation frequency

2. OpAmplIdeal is a noiseless component.
3. The recommendation is to use a Gain value no greater than 1e10, or 200 dB for practical purposes.

**Equivalent Circuit**

![Equivalent Circuit Diagram]

```
1 + \text{V}^- \rightarrow Z1 \rightarrow Z4 \rightarrow Z2 + 4 - \text{V}^+ \rightarrow Z3 + \text{Vs} \rightarrow 5
```

OpAmpIdeal (Ideal Operational Amplifier) 1-57
VMult (Voltage Multiplier)

Symbol

Parameters

R1 = reference resistance for port 1, in ohms
R2 = reference resistance for port 2, in ohms
R3 = reference resistance for port 3, in ohms
L31 = loss in dB, pin 1 to pin 3; for linear analysis only
L32 = loss in dB, pin 2 to pin 3; for linear analysis only
Linear = yes enables linear analysis mode, for use with linear simulations

Notes/Equations

1. VMult uses reference input and output impedances. Its output voltage is equal to the product of its two input voltages.

2. If Linear is set to yes,

\[ V_3 = \frac{A_{31}V_1 + A_{32}V_2}{R_3} \]

A31 and A32 are the L31 and L32 losses converted from dB to regular values.

If Linear is set to no, V3 is formed from V1 and V2 through an ideal mixing process; for information on this process refer to the Ideal Mixer in the SDD Examples section in Chapter 5 in the User-Defined Models manual.
Chapter 2: Filters

Introduction

The Filters - <filter type> and System - <device type> palettes contain two fundamentally different types of behavioral system models.

Filters, System - Amps & Mixers, and System - Mod/ Demod can be classified as tops-down system models that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as parameter-based behavioral models.

System - Data Models can be classified as bottoms-up system models that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as data-based behavioral models.

The parameter-based behavioral models typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The data-based behavioral models typically provide superior accuracy relative to the parameter-based behavioral models as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between parameter- and data-based behavioral models justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, demodulators etc. grouped together).

The use model for parameter-based behavioral models is to simply set a series of parameters prior to using the model. The use model for data-based behavioral models is described in Chapter 8, System Data Models.

Filter Categories

The filter component libraries contain filters in eight response categories: Butterworth, Chebyshev, Elliptic, Gaussian, Bessel-Thompson, Raised-Cosine, Pole-Zero, and Polynomial. Lowpass, highpass, bandpass, and bandstop filters are available in each category.
Butterworth, Chebyshev, and Elliptic filters have good selectivity but poor group delay flatness; Bessel-Thompson and Gaussian filters have good delay flatness but poor selectivity. Raised-Cosine filters are uni-directional ideal Nyquist filters for bandlimiting digital signals. Pole-Zero and Polynomial filters allow users to define arbitrary response shapes.

Except for the Raised-Cosine category, the filter S-parameters are calculated based on standard filter synthesis theory [1]. $S_{21}$ and $S_{12}$ include losses specified by unloaded quality factor ($Q_u$) and insertion loss IL, where applicable. The assumption is made that filter pole predistortion is used to preserve the specified frequency response in the presence of losses [2]. However, $S_{11}$ and $S_{22}$ neglect losses, an approximation that causes little error for realizable filters.

The basic nature of the response of a lowpass, highpass, bandpass, and bandstop filter is illustrated below. The illustrations include certain filter parameter definitions: $f_{\text{pass}}$, $f_{\text{stop}}$, $A_{\text{pass}}$, and $A_{\text{stop}}$ for lowpass/highpass filters, and $f_{\text{center}}$, $BW_{\text{pass}}$, $BW_{\text{stop}}$, $A_{\text{pass}}$, and $A_{\text{stop}}$ for bandpass/bandstop filters. (Note that all filters do not have all of these parameters.)

In addition to these filter parameters, Gaussian and Bessel filters (XXX_Gaussian and XXX_Bessel (XXX = LPF, HPF, BPF, or BSF) have a group delay parameter $GD_{\text{pass}}$. This parameter is motivated by the fact that a signal experiences a delay when passing through a filter. Calculated as the negative of the derivative of the phase response with respect to frequency, this delay (the group delay) will be frequency dependent for filters with a non-linear phase response. In other words, the group delay at a given frequency specifies the delay experienced by a group of sinusoidal components all having frequencies within a narrow interval around that frequency. A filter whose time-domain impulse response is symmetric around $t=0$ must be subjected to a certain finite shift (the group delay) in order to maintain (at least some semblance of) causality when a transient simulator truncates its impulse response below zero. Different time-frequency characteristics for different filters can lead to different requirements for this group delay.

For LPF_Bessel and LPF_Gaussian, the group delay parameter $GD_{\text{pass}}$ is defined as the group delay at the passband edge frequency relative to that at zero frequency (this is illustrated below). For highpass, bandpass, or bandstop filters, the definition is the same except that the group delay at infinity (highpass) or at the center frequency (bandpass, bandstop) is used as the reference. $GD_{\text{pass}}$ cannot drop below zero and cannot exceed one; its default value is 0.9. The group delay value has a significant effect on filter order and therefore filter rejection. Larger $GD_{\text{pass}}$ values...
will result in longer delays and larger filter orders; smaller $GD_{\text{pass}}$ values will result in smaller delays and smaller filter orders. If your Gaussian or Bessel filter provides less out-of-band rejection than you expect, try increasing the $GD_{\text{pass}}$ parameter.

There are two ways of dealing with filter order, $N$:

- Leave the filter order at zero and specify the parameters characterizing the behavior of the filter, per the illustrations below. Given these specifications, the program will calculate and report filter order $N$ which meets these specifications. Because $N$ must be an integer, the calculated order $N$ exceeds the specifications in most cases. $N$ is capped at the upper value of 15. For Chebyshev filters, the calculated filter order $N$ must be an odd number to ensure filter symmetry.

- Alternatively, specify filter order $N$ explicitly. If a non-zero $N$ is specified, it will overwrite the filter specifications. The filter response is simply calculated based on the specified order $N$. As above, $N$ must be an integer and will be capped at 15. And, $N$ must be odd for Chebyshev filters.

\[ GD(f) \]
\[ GD_{\text{pass}} = \frac{T_{\text{pass}}}{T_{0}} \]

![Graph](image-url)
Lowpass Filter Behavior at DC

At DC, a lowpass filter appears to reduce to a wire. During DC analysis, one would therefore intuitively expect the S-matrix of a lowpass filter to reduce to $S_{11}=S_{22}=1$ and $S_{12}=S_{21}=0$. Also, one would intuitively expect the input and output voltages and currents to fulfill $v_2=v_1$ and $i_2=i_1$. For $Z_1=Z_2$, both these expectations are met. However, if $Z_1$ and $Z_2$ are unequal, there is simply no way to mathematically realize a meaningful filter that meets both expectations. If it meets the expectations in the S-domain, it won't for voltages and currents and vice versa.

The ADS lowpass filters are implemented in the S-domain, not the voltage/current-domain. At DC, the lowpass filters therefore reduce to a perfect $[1 \ 0; 0 \ 1]$ S-matrix regardless of the choice of reference impedance. The output voltage and current, however, are given by $v_2=\sqrt{Z_2/Z_1} \times v_1$ and $i_2=\sqrt{Z_1/Z_2} \times i_1$. This, as pointed out previously, reduces to $v_2=v_1$ and $i_2=i_1$ only for $Z_1=Z_2$. These relations can easily be derived from power conservation and can also be found from more rigorous S-parameter analysis.
References


BPF_Bessel (Bandpass Filter, Bessel-Thompson)

Symbol

Parameters
- Fcenter = center frequency, in hertz
- BWpass = width measured from lower to upper passband edges, in hertz
- Apass = attenuation at passband edges, in dB
- GDpass = group delay at passband edges relative to that at center frequency
- StopType = stopband input impedance type: OPEN or SHORT
- MaxRej = maximum rejection level, in dB
- N = filter order; if not given, it is calculated based on BWpass, Apass, and GDpass
- IL = insertion loss, in dB
- Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
- Z1 = reference impedance for port 1, in ohms
- Z2 = reference impedance for port 2, in ohms
- Temp = temperature in °C

Range of Usage
- BWpass < Fcenter
- 0.01 ≤ Apass ≤ 3.0
- 0 < GDpass < 1
- 1 ≤ N ≤ 15
- Qu ≥ 1

Notes/Equations
1. Refer to “Filter Categories” on page 2-1 at the beginning of this chapter.
2. This component has no default artwork associated with it.
BPF_Butterworth (Bandpass Filter, Butterworth)

Symbol

Parameters
F\text{center} = \text{center frequency, in hertz}
BW\text{pass} = \text{width of passband, in hertz}
Apass = \text{attenuation at passband edges, in dB}
BW\text{stop} = \text{width measured from lower to upper stopband edges, in hertz}
Astop = \text{attenuation at stopband edges, in dB}
StopType = \text{stopband input impedance type: OPEN or SHORT}
Max\text{Rej} = \text{maximum rejection level, in dB}
N = \text{filter order; if not given, it is calculated based on } BW\text{pass}, \text{Apass}, \text{BW\text{stop}}, \text{and } A\text{stop}
IL = \text{insertion loss, in dB}
Qu = \text{unloaded quality factor for resonators, default setting is an infinite } Qu \text{ and expresses a dissipationless resonant circuit.}
Z1 = \text{reference impedance for port 1, in ohms}
Z2 = \text{reference impedance for port 2, in ohms}
Temp = \text{temperature in } ^\circ\text{C}

Range of Usage
1 \leq N \leq 15
BW\text{pass} < F\text{center}
0.01 \leq \text{Apass} \leq 3.0
Qu \geq 1

Notes/Equations
1. Refer to "Filter Categories" on page 2-1.
2. This component has no default artwork associated with it.
Filters

BPF_Chebyshev (Bandpass Filter, Chebyshev)

Symbol

Parameters

Fcenter = center frequency, in hertz
BWpass = width of passband, in hertz
Apass = attenuation at passband edges, in dB; typically Apass=Ripple
Ripple = passband ripple, in dB
BWstop = width measured from lower to upper stopband edges, in hertz
Astop = attenuation at stopband edges, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, is calculated based on BWpass, Ripple, BWstop, and Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms
Temp = temperature in °C

Range of Usage

BWpass < Fcenter
0.01 ≤ Ripple ≤ 3.0
1 ≤ N ≤ 15
Qu ≥ 1
Notes/Equations

1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BPF_Elliptic (Bandpass Filter, Elliptic)

Symbol

Parameters
Fcenter = center frequency, in hertz
BWpass = width of passband, in hertz
Ripple = passband ripple, in dB
BWstop = width measured from lower to upper stopband edges, in hertz
Astop = attenuation at stopband edges, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, is calculated based on BWpass, Ripple, BWstop, and Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms
Temp = temperature in °C

Range of Usage
BWpass < Fcenter
0.01 \leq \text{Ripple} \leq 3.0
Astop > 0
1 \leq N \leq 15

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BPF_Gaussian (Bandpass Filter, Gaussian)

Symbol

Parameters
- **Fcenter** = center frequency, in hertz
- **BWpass** = width measured from lower to upper passband edges, in hertz
- **Apass** = attenuation at passband edges, in dB
- **GDpass** = group delay at passband edges relative to that at center frequency
- **StopType** = stopband input impedance type: OPEN or SHORT
- **MaxRej** = maximum rejection level, in dB
- **N** = filter order; if not given, it is calculated based on BWpass, Apass and GDpass
- **IL** = insertion loss, in dB
- **Qu** = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
- **Z1** = reference impedance for port 1, in ohms
- **Z2** = reference impedance for port 2, in ohms
- **Temp** = temperature in °C

Range of Usage
- \( BWpass < Fcenter \)
- \( 0.01 \leq Apass \leq 3.0 \)
- \( 0 < GDpass < 1 \)
- \( 1 \leq N \leq 15 \)
- \( Qu \geq 1 \)

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BPF_PoleZero (Bandpass Filter, Pole Zero)

Symbol

Parameters
Poles = list of poles
Zeros = list of zeros
Gain = gain factor
Fcenter = center frequency, frequency unit
BWpass = 3dB bandwidth, frequency unit
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations
1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.
   The transfer function for the filter is:

   \[ S_{21} = \text{Gain} \left( \frac{(S - \text{Zero1})(S - \text{Zero2})\ldots}{(S - \text{Pole1})(S - \text{Pole2})\ldots} \right) \]

   where

   \[ S = j \times \frac{\text{Fcenter}}{\text{BWpass}} \left( \frac{\text{Freq}}{\text{Fcenter}} - \frac{\text{Fcenter}}{\text{Freq}} \right) \]

   and

   Freq is the analysis frequency
   At least one pole must be supplied.
3. The following example demonstrates interpretation of simulation results with this component. From the user-specified poles/zeros, we derive:

\[ S_{21,\text{Lowpass Prototype}} = \text{Gain} \times \frac{(s-Z_1) \times \cdots \times (s-Z_n)}{(s-P_1) \times \cdots \times (s-P_m)} \]

We then check to see if \( S_{21,\text{Lowpass Prototype}} > 1 \). If yes, we scale \( S_{21} \) by another factor to make sure \( S_{21,\text{Max}} \leq 1 \). We then derive \( S_{11} \) (\( S_{22} \)) from the following formula:

\[ S_{11}^2 + S_{21}^2 = 1 \]

In this example, when \( \text{Gain} \) is set > 0.471151, then \( S_{11} \) is derived as what you will expect. If \( \text{Gain} \) in your example is < 0.471151, then \( S_{11} \) derived from the preceding equation, will be much higher that what you will expect. In this situation, set \( \text{Gain} \) to be 0.1 so that \( S_{21} \) has a lot of insertion loss. But we assumed there is no insertion loss in deriving \( S_{11} \).

There are other alternatives:

- Use \( S_{2P,\text{Eqn}} \) so that you can define the \( S_{21} \) and \( S_{11} \) polynomials however you want. You can define this as follows:
  \[ s=j\omega, \ S_{21} = \text{Gain} \times (s-Z_1) \times \cdots \times (s-Z_n) / (s-P_1) \times \cdots \times (s-P_n), \ S_{11}=\text{<your choice>} \]
- Use \( BPF\_\text{Pole}_\text{Zero} \) to model a lossless BPF, then use an attenuator to add insertion loss.
BPF_Polynomial (Bandpass Filter, Polynomial)

Symbol

Parameters
Denominator = denominator coefficients
Numerator = numerator coefficients
Gain = gain factor
Fcenter = center frequency, in hertz
BWpass = 3dB bandwidth, in hertz
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations
1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients.
   The transfer function for the filter is:
   \[
   S_{21} = \frac{Gain \left( N_0 + (N_1 \times S) + (N_2 \times S^2)\ldots \right)}{D_0 + (D_1 \times S) + (D_2 \times S^2)\ldots} \]

   where
   \[
   S = j \times \frac{(Freq/F_o - F_o/Freq)/(F_{high}/F_o = F_o/F_{high})}{F_{high} = F_{center} + 0.5 \times BWpass}
   \]
   and
   \[
   F_o = \sqrt{(F_{center} - 0.5 \times BWpass) \times (F_{center} + 0.5 \times BWpass))}
   \]
BPF_RaisedCos (Bandpass Filter, Raised-Cosine)

Symbol

Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35
Fcenter = Center frequency, in Hz. Default = 1.5 GHz.
SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 kHz
DelaySymbols = Number of symbols delayed by filter. Default = 5
Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5
DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0
SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no
Gain = Gain normalization factor. Default = 1.0
Zout = Output impedance, in Ohms. Default = 50 Ohms
WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming
ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.
ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.
ImpMaxPts = Maximum number of points in impulse response. Default = 4096
Other = output string to netlist

Range of Usage

0 ≤ Alpha ≤ 1
DelaySymbols ≥ 1
0 ≤ Exponent ≤ 1
0 ≤ DutyCycle ≤ 100
Notes/Equations

1. Refer to “Filter Categories” on page 2-1.

2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Zout.

3. The voltage gain is described by the following function.

\[
\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{\text{Gain} \times G_{\text{filt}}}{G_{\text{comp}}} e^{-j2\pi \text{frequency}(\text{DelaySymbols}/\text{SymbolRate})}
\]

where

\[
G_{\text{filt}} = \begin{cases} 
1 & \text{for } \text{Freq} < 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
0 & \text{for } \text{Freq} > 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate}
\end{cases}
\]

\[
[0.5(1-\sin(\pi \times (\text{Freq} - 0.5 \times \text{SymbolRate})/(\text{Alpha} \times \text{SymbolRate})))^{\text{Exponent}}
\]

\[
G_{\text{comp}} = \begin{cases} 
1.0 & \text{if DutyCycle =0, else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)]^{\text{Exponent}} & \text{if SincE =YES, else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)] & \text{if SincE =NO}
\end{cases}
\]

\[
\text{Freq} = \text{abs(\text{Fcenter} - \text{frequency})}
\]

\[
\text{sinc}(x) = \frac{\sin(x)}{x}
\]

\[
x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency}/\text{SymbolRate}
\]

4. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.

5. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of −90dBc or smaller.
6. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.

7. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp}. (Note that the Exponent term is always present in the G_{filt} term.)

8. This component has no default artwork associated with it.
BSF_Bessel (Bandstop Filter, Bessel-Thompson)

Symbol

Parameters
Fcenter = center frequency, in hertz
BWpass = width measured from lower to upper passband edges, in hertz
Apass = attenuation at passband edges, in dB
GDpass = group delay at passband edges relative to that at center frequency
StopType = stopband input impedance type: OPEN or SHORT
Maxrej = maximum rejection level, in dB
N = filter order; if not given, it is calculated based on BWpass, Apass and GDpass
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage
BWpass < Fcenter
0.01 ≤ Apass ≤ 3.0
0 < GDpass < 1
1 ≤ N ≤ 15

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BSF_Butterworth (Bandstop Filter, Butterworth)

Symbol

Parameters
Fcenter = center frequency, in hertz
BWstop = width of stop band, in hertz
Astop = attenuation at stopband edges, in dB
BWpass = width measured from lower to upper passband edges, in hertz
Apass = attenuation at passband edges, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, it is calculated based on BWpass, Apass, BWstop, Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage
BWpass < Fcenter
0.01 ≤ Apass ≤ 3.0
1 ≤ N ≤ 15

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BSF_Chebyshev (Bandstop Filter, Chebyshev)

Symbol

Parameters

- $F_{\text{center}}$ = center frequency, in hertz
- $BW_{\text{stop}}$ = width of stop band, in hertz
- $A_{\text{stop}}$ = attenuation at stopband edges, in dB
- $\text{Ripple}$ = stopband ripple, in dB
- $BW_{\text{pass}}$ = width measured from lower to upper passband edges, in hertz
- $A_{\text{pass}}$ = attenuation at passband edges, in dB
- $\text{StopType}$ = stopband input impedance type: OPEN or SHORT
- $\text{MaxRej}$ = maximum rejection level, in dB
- $N$ = filter order; if not given, it is calculated based on $BW_{\text{pass}}$, $A_{\text{pass}}$, $BW_{\text{stop}}$, $A_{\text{stop}}$
- $IL$ = insertion loss, in dB
- $Qu$ = unloaded quality factor for resonators, default setting is an infinite $Qu$ and expresses a dissipationless resonant circuit.
- $Z1$ = reference impedance for port 1, in ohms
- $Z2$ = reference impedance for port 2, in ohm
- $\text{Temp}$ = temperature, in °C

Range of Usage

- $BW_{\text{pass}} < F_{\text{center}}$
- $0.01 \leq \text{Ripple} \leq 3.0$
- $1 \leq N \leq 15$

Notes/Equations

1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BSF_Elliptic (Bandstop Filter, Elliptic)

Symbol

Parameters

Fcenter = center frequency, in hertz
BWstop = width of stop band, in hertz
Astop = attenuation at stopband edges, in dB
Ripple = stopband ripple, in dB
BWpass = width measured from lower to upper passband edges, in hertz
Apass = attenuation at passband edges, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, is calculated based on BWpass, Apass, BWstop, Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage

BWpass < Fcenter
0.01 ≤ Ripple ≤ 3.0
1 ≤ N ≤ 15

Notes/Equations

1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
Filters

**BSF_Gaussian (Bandstop Filter, Gaussian)**

**Parameters**
- Fcenter = center frequency, in hertz
- BWpass = width measured from lower to upper passband edges, in hertz
- Apass = attenuation at passband edges, in dB
- GDpass = group delay at passband edges relative to that at center frequency
- StopType = stopband input impedance type: OPEN or SHORT
- MaxRej = maximum rejection level, in dB
- N = filter order; if not given, it is calculated based on BWpass, Apass and GDpass
- IL = insertion loss, in dB
- Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
- Z1 = reference impedance for port 1, in ohms
- Z2 = reference impedance for port 2, in ohm
- Temp = temperature, in °C

**Range of Usage**
- BWpass < Fcenter
- 0.01 ≤ Apass ≤ 3.0
- 0 < GDpass < 1
- 1 ≤ N ≤ 15

**Notes/Equations**
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
BSF_PoleZero (Bandstop Filter, Pole Zero)

Symbol

Parameters
Poles = list of poles
Zeros = list of zeros
Gain = gain factor
Fcenter = center frequency, in hertz
BWpass = 3dB bandwidth of lower to upper passband edges, in hertz
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations
1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

\[
S_{21} = \text{Gain} \frac{(S - \text{Zero}_1)(S - \text{Zero}_2)\ldots}{(S - \text{Pole}_1)(S - \text{Pole}_2)\ldots}
\]

where

\[
S = j \left( \frac{\text{Fcenter}}{\text{BWpass}} \right) \times \left( \frac{\text{Freq}}{\text{Fcenter}} - \frac{\text{Fcenter}}{\text{Freq}} \right)
\]

and

Freq is the analysis frequency
At least one pole must be supplied.
BSF_Polynomial (Bandstop Filter, Polynomial)

Symbol

![Symbol Image]

Parameters

Denominator = list of denominator coefficients
Numerator = list of numerator coefficients
Gain = gain factor
Fcenter = center frequency, in hertz
BWpass = width of bandpass, in hertz
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Denominator and Numerator are lists of polynomial coefficients.

The transfer function for the filter is:

\[
S_{21} = \frac{Gain \left( N_0 + (N_1 \times S) + (N_2 \times S^2) + \ldots \right)}{D_0 + (D_1 \times S) + (D_2 \times S^2) + \ldots}
\]

where

\[
S = -j \times \left( F_0/F_{low} - F_{low}/F_0 \right)/(F_{req}/F_0 - F_0/F_{req})
\]

and

\[
F_{req} \text{ is the analysis frequency}
F_{low} = F_{center} - 0.5 \times BWpass
F_0 = \sqrt{\left( F_{center} - 0.5 \times BWpass \right) \times F_{center} + 0.5 \times BWpass})
\]

At least one Denominator coefficient must be supplied.
**BSF_RaisedCos (Bandstop Filter, Raised-Cosine)**

**Symbol**

![Symbol](image)

**Parameters**

- **Alpha** = Roll off factor defining filter excess bandwidth. Default = 0.35
- **Fcenter** = Center frequency, in Hz. Default = 1.5 GHz.
- **SymbolRate** = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 kHz
- **DelaySymbols** = Number of symbols delayed by filter. Default = 5
- **Exponent** = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5
- **DutyCycle** = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0
- **SincE** = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no
- **Gain** = Gain normalization factor. Default = 1.0
- **Zout** = Output impedance, in Ohms. Default = 50 Ohms
- **WindowType** = Window type applied to impulse response. 0=None (default), 1=Hann, 2=Hamming
- **ImpMaxFreq** = Maximum frequency to consider when calculating impulse response, in Hz.
- **ImpDeltaFreq** = Frequency sample spacing when calculating impulse response, in Hz.
- **ImpMaxPts** = Maximum number of points in impulse response. Default = 4096
- **Other** = output string to netlist

**Range of Usage**

- \( 0 \leq \text{Alpha} \leq 1 \)
- \( \text{DelaySymbols} \geq 1 \)
- \( 0 \leq \text{Exponent} \leq 1 \)
- \( 0 \leq \text{DutyCycle} \leq 100 \)
Notes/Equations

1. Refer to “Filter Categories” on page 2-1.

2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Zout.

3. The voltage gain is described by the following function.

\[
\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt} \times e^{-j2\pi frequency (DelaySymbols/symbolRate)}}{G_{comp}}
\]

where

\[G_{filt} = \begin{cases} 
0 & \text{for } \text{Freq} < 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
1 & \text{for } \text{Freq} > 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
[0.5(1+\sin(\pi \times (\text{Freq}-0.5 \times \text{SymbolRate})/(\text{Alpha} \times \text{SymbolRate})))]^{\text{Exponent}} & \text{otherwise}
\end{cases}\]

\[G_{comp} = \begin{cases} 
1.0 & \text{if DutyCycle } = 0, \text{ else } \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)]^{\text{Exponent}} & \text{if SincE } = \text{YES, else } \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)] & \text{if SincE } = \text{NO}
\end{cases}\]

\[\text{Freq} = \text{abs (Fcenter - frequency)}\]

\[\text{sinc}(x) = \frac{\sin(x)}{x}\]

\[x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency}/\text{SymbolRate}\]

4. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.

5. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of −90dBc or smaller.
6. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.

7. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE = YES, Exponent will be applied to the gain compensation term, G_{comp}. (Note that the Exponent term is always present in the G_{filt} term.)

8. This component has no default artwork associated with it.
HPF_Bessel (Highpass Filter, Bessel-Thompson)

Symbol

Parameters
Fpass = passband edge frequency, in hertz
Apass = attenuation at passband edge frequency, in dB
GDpass = group delay at passband edge frequency relative to that at infinite frequency
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, it is calculated based on Fpass, Apass and GDpass
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage
Fpass > 0
0.01 ≤ Apass ≤ 3.0
0 ≤ GDpass < 1
1 ≤ N ≤ 15

Notes/Eqautions
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
HPF_Butterworth (Highpass Filter, Butterworth)

Symbol

Parameters
Fpass = passband edge frequency, in hertz
Apass = attenuation at passband edge frequency, in dB
Fstop = stopband edge frequency, in hertz
Astop = attenuation at stopband edge frequency, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, it is calculated based on Fpass, Apass, Fstop, and Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage
Fpass > 0
0.01 ≤ Apass ≤ 3.0
1 ≤ N ≤ 15

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
Filters

HPF_Chebyshev (Highpass Filter, Chebyshev)
Symbol

Parameters
Fpass = passband edge frequency, in hertz
Apass = attenuation at passband edge frequency, in dB. Typically, Apass=Ripple
Ripple = passband ripple, in dB
Fstop = stopband edge frequency, in hertz
Astop = attenuation at stopband edge frequency, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, is calculated based on Fpass, Ripple, Fstop, and Astop
IL = insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage
Fpass > 0
0.01 ≤ Ripple ≤ 3.0
1 ≤ N ≤ 15

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
**HPF_Elliptic (Highpass Filter, Elliptic)**

**Symbol**

![Symbol Image]

**Parameters**

- $F_{\text{pass}} =$ passband edge frequency, in hertz
- Ripple = passband ripple, in dB
- $F_{\text{stop}} =$ stopband edge frequency, in hertz
- $A_{\text{stop}} =$ attenuation at stopband edge frequency, in dB
- StopType = stopband input impedance type: OPEN or SHORT
- $\text{MaxRej} =$ maximum rejection level, in dB
- $N =$ filter order; if not given, is calculated based on $F_{\text{pass}}, \text{Ripple}, F_{\text{stop}}, \text{and} A_{\text{stop}}$
- $\text{IL} =$ insertion loss, in dB
- $\text{Qu} =$ unloaded quality factor for resonators, default setting is an infinite $\text{Qu}$ and expresses a dissipationless resonant circuit.
- $Z1 =$ reference impedance for port 1, in ohms
- $Z2 =$ reference impedance for port 2, in ohm
- Temp = temperature, in °C

**Range of Usage**

- $F_{\text{pass}} > 0$
- $0.01 \leq \text{Ripple} \leq 3.0$
- $A_{\text{stop}} > 0$
- $1 \leq N \leq 15$

**Notes/Equations**

1. Refer to "Filter Categories" on page 2-1.
2. This component has no default artwork associated with it.
Filters

**HPF_Gaussian (Highpass Filter, Gaussian)**

**Symbol**

![Symbol](image)

**Parameters**

- **Fpass** = passband edge frequency, in hertz
- **Apass** = attenuation at passband edge frequency, in dB
- **GDpass** = group delay at passband edge frequency relative to that at infinite frequency
- **StopType** = stopband input impedance type: OPEN or SHORT
- **MaxRej** = maximum rejection level, in dB
- **N** = filter order; if not given, it is calculated based on **Fpass**, **Apass** and **GDpass**
- **IL** = insertion loss, in dB
- **Qu** = unloaded quality factor for resonators, default setting is an infinite **Qu** and expresses a dissipationless resonant circuit.
- **Z1** = reference impedance for port 1, in ohms
- **Z2** = reference impedance for port 2, in ohm
- **Temp** = temperature, in °C

**Range of Usage**

- **Fpass** > 0
- 0.01 ≤ **Apass** ≤ 3.0
- 0 < **GDpass** < 1
- 1 ≤ **N** ≤ 15

**Notes/Equations**

1. Refer to “Filter Categories” on page 2-1.
2. This component has no default artwork associated with it.
HPF_PoleZero (Highpass Filter, Pole Zero)

Symbol

Parameters

Poles = list of poles
Zeros = list of zeros
Gain = gain factor
Fpass = 3dB passband edge frequency, in hertz
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations

1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.

The transfer function for the filter is:

$$S_{21} = \frac{Gain(S - Zero_1)(S - Zero_2)\ldots}{(S - Pole_1)(S - Pole_2)\ldots}$$

where

$$S = j(Fpass/Freq)$$

and

FREQ is the analysis frequency
At least one pole must be supplied.
Filters

**HPF_Polynomial (Highpass Filter, Polynomial)**

**Symbol**

![Symbol](image)

**Parameters**
- Numerator = list of numerator coefficients
- Denominator = list of denominator coefficients
- Gain = gain factor
- Fpass = 3dB passband edge frequency, in hertz
- StopType = stopband input impedance type: OPEN or SHORT
- Z1 = reference impedance for port 1, in ohms
- Z2 = reference impedance for port 2, in ohms

**Notes/Equations**
1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients.
   - The transfer function for the filter is:
     \[
     S_{21} = \frac{Gain \left( N_0 + (N_1 \times S) + (N_2 \times S^2) \ldots \right)}{D_0 + (D_1 \times S) + (D_2 \times S^2) \ldots}
     \]
   - where
     \[ S = -j(Fpass/Freq) \]
   - and
     \[ Freq \text{ is the analysis frequency} \]
### HPF_RaisedCos (Highpass Filter, Raised-Cosine)

#### Symbol

![Symbol](image)

#### Parameters

- **Alpha** = Roll off factor defining filter excess bandwidth. Default = 0.35
- **SymbolRate** = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 KHz
- **DelaySymbols** = Number of symbols delayed by filter. Default = 5
- **Exponent** = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5
- **DutyCycle** = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0
- **SincE** = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no
- **Gain** = Gain normalization factor. Default = 1.0
- **Zout** = Output impedance, in Ohms. Default = 50 Ohms
- **WindowType** = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming
- **ImpMaxFreq** = Maximum frequency to consider when calculating impulse response, in Hz.
- **ImpDeltaFreq** = Frequency sample spacing when calculating impulse response, in Hz.
- **ImpMaxPts** = Maximum number of points in impulse response. Default = 4096
- **Other** = output string to netlist

#### Range of Usage

- $0 \leq \text{Alpha} \leq 1$
- $\text{DelaySymbols} \geq 1$
- $0 \leq \text{Exponent} \leq 1$
- $0 \leq \text{DutyCycle} \leq 100$

#### Notes/Equations
Filters

1. Refer to "Filter Categories" on page 2-1.

2. This filter is unidirectional; input impedance is infinite; output impedance is specified by Zout.

3. This filter is unidirectional; input impedance is infinite; output impedance is specified by Zout.

4. The voltage gain is described by the following function.

\[
\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{\text{Gain} \times G_{\text{filt}}}{G_{\text{comp}}} e^{j2\pi\text{frequency}(\text{DelaySymbols}/\text{SymbolRate})}
\]

where

\[
G_{\text{filt}} = \begin{cases} 
0 & \text{for frequency} < 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
1 & \text{for frequency} > 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
[0.5(1+\sin(\pi\times(\text{frequency}-0.5\times\text{SymbolRate})/(\text{Alpha}\times\text{SymbolRate})))]^{\text{Exponent}} & \text{otherwise}
\end{cases}
\]

\[
G_{\text{comp}} = \begin{cases} 
1.0 & \text{if DutyCycle} = 0, \text{else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc(x)}]^{\text{Exponent}} & \text{if SincE = YES, else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc(x)}] & \text{if SincE = NO}
\end{cases}
\]

\[
\text{Freq} = \text{abs} (F_{\text{center}} - \text{frequency})
\]

\[
\text{sinc(x)} = \sin(x)/x
\]

\[
x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency}/\text{SymbolRate}
\]

5. While the Exponent can be any value, the two standard values are 1.0 for the ideal Raised-Cosine filter response or 0.5 to simulate the Root Raised-Cosine filter response when present at both the receiving and transmitting channels.

6. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.
Using a Hanning window (WindowType=1) a DelaySymbols parameter of 15 should result in equivalent frequency domain sidelobes of −90dBc or smaller.

7. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.

8. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE=YES, Exponent will be applied to the gain compensation term, G_{comp}. (Note that the Exponent term is always present in the G_{filt} term.)

9. This component has no default artwork associated with it.
LPF_Bessel (Lowpass Filter, Bessel-Thompson)

Symbol

Parameters
- \( F_{\text{pass}} \) = passband edge frequency, in hertz
- \( A_{\text{pass}} \) = attenuation at passband edge frequency, in dB
- \( G_{\text{Dpass}} \) = group delay at passband edge frequency relative to that at zero frequency
- \( \text{StopType} \) = stopband input impedance type: OPEN or SHORT
- \( \text{MaxRej} \) = maximum rejection level, in dB
- \( N \) = filter order; if not given, it is calculated based on \( F_{\text{pass}}, A_{\text{pass}}, \) and \( G_{\text{Dpass}} \)
- \( \text{IL} \) = passband insertion loss, in dB
- \( \text{Qu} \) = unloaded quality factor for resonators, default setting is an infinite \( \text{Qu} \) and expresses a dissipationless resonant circuit.
- \( Z_1 \) = reference impedance for port 1, in ohms
- \( Z_2 \) = reference impedance for port 2, in ohm
- \( \text{Temp} \) = temperature, in °C

Range of Usage
- \( F_{\text{pass}} > 0 \)
- \( 0.01 \leq A_{\text{pass}} \leq 3.0 \)
- \( 0 < G_{\text{Dpass}} < 1 \)
- \( 1 \leq N \leq 15 \)
- \( \text{Qu} \geq 1 \)

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. For information on LPF behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
3. This component has no default artwork associated with it.
LPF_Butterworth (Lowpass Filter, Butterworth)

Symbol

Parameters
- Fpass = passband edge frequency, in hertz
- Apass = attenuation at passband edge, in dB
- Fstop = stopband edge frequency, in hertz
- Astop = attenuation at stopband edge, in dB
- StopType = stopband input impedance type: OPEN or SHORT
- MaxRej = maximum rejection level, in dB
- N = filter order (if N > 0, it overwrites Fstop and Astop)
- IL = passband insertion loss, in dB
- Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
- Z1 = reference impedance for port 1, in ohms
- Z2 = reference impedance for port 2, in ohm
- Temp = temperature, in °C

Range of Usage
- Fpass > 0
- 0.01 ≤ Apass ≤ 3.0
- 1 ≤ N ≤ 15
- Qu ≥ 1

Notes/Equations
1. Refer to “Filter Categories” on page 2-1.
2. For information on LPF behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
3. This component has no default artwork associated with it.
Filters

**LPF_Chebyshev (Lowpass Filter, Chebyshev)**

**Symbol**

![Symbol Image]  

**Parameters**

- **Fpass** = passband edge frequency, in hertz
- **Apass** = attenuation at passband edge, in dB. By default, Apass=Ripple
- **Ripple** = passband ripple, in dB
- **Fstop** = stopband edge frequency, in hertz
- **Astop** = attenuation at stopband edge, in dB
- **StopType** = stopband input impedance type: OPEN or SHORT
- **MaxRej** = maximum rejection level, in dB
- **N** = filter order (if N > 0, it overwrites Fstop, and Astop)
- **IL** = passband insertion loss, in dB
- **Qu** = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
- **Z1** = reference impedance for port 1, in ohms
- **Z2** = reference impedance for port 2, in ohm
- **Temp** = temperature, in °C

**Range of Usage**

- **Fpass** > 0
- **0.01 ≤ Ripple ≤ 3.0**
- **1 ≤ N ≤ 15**
- **Qu ≥ 1**

**Notes/Equations**

1. Refer to "Filter Categories" on page 2-1.
2. For information on LPF behavior at DC, refer to "Lowpass Filter Behavior at DC" on page 2-4.
3. This component has no default artwork associated with it.
LPF_Elliptic (Lowpass Filter, Elliptic)

Symbol

Parameters

Fpass = passband edge frequency, in hertz
Ripple = passband ripple, in dB
Fstop = stopband edge frequency, in hertz
Astop = attenuation at stopband edge, in dB
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order (if N > 0, it overwrites Fstop, and Astop)
IL = passband insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage

Fpass > 0
0.01 ≤ Ripple ≤ 3.0
Astop > 0
1 ≤ N ≤ 15

Notes/Equations

1. Refer to “Filter Categories” on page 2-1.
2. For information on LPF behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
3. This component has no default artwork associated with it.
LPF_Gaussian (Lowpass Filter, Gaussian)

Symbol

Parameters

Fpass = passband edge frequency, in hertz
Apass = attenuation at passband edge frequency, in dB
GDpass = group delay at passband edge frequency relative to that at zero frequency
StopType = stopband input impedance type: OPEN or SHORT
MaxRej = maximum rejection level, in dB
N = filter order; if not given, it is calculated based on Fpass, Apass, and GDpass
IL = passband insertion loss, in dB
Qu = unloaded quality factor for resonators, default setting is an infinite Qu and expresses a dissipationless resonant circuit.
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohm
Temp = temperature, in °C

Range of Usage

Fpass > 0
0.01 ≤ Apass ≤ 3.0
0 < GDpass < 1
1 ≤ N ≤ 15
Qu ≥ 1

Notes/Equations

1. Refer to “Filter Categories” on page 2-1.
2. For information on LPF behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
3. This component has no default artwork associated with it.
LPF_GMSK (Lowpass Filter, GMSK)

Symbol

Parameters

\[ BT = \text{Product of 3dB bandwidth and bit duration} \]
\[ \text{BitRate} = \text{Digital bit rate defining filter bandwidth, in Hz. Default = 270.833 KHz} \]
\[ \text{DelayBits} = \text{Number of bits delayed by filter. Default = 5} \]
\[ \text{Gain} = \text{Gain normalization factor. Default = 1.0} \]
\[ Z_{out} = \text{Output impedance, in Ohms. Default = 50 Ohms} \]
\[ \text{WindowType} = \text{Window type applied to impulse response. 0=None (default), 1=Hann, 2=Hamming} \]
\[ ImpMaxFreq = \text{Maximum frequency to consider when calculating impulse response, in Hz.} \]
\[ ImpDeltaFreq = \text{Frequency sample spacing when calculating impulse response, in Hz.} \]
\[ ImpMaxPts = \text{Maximum number of points in impulse response. Default = 4096} \]

Range of Usage

\[ 0 \leq \text{Alpha} \leq 1 \]
\[ \text{DelayBits} \geq 1 \]

Notes/Equations/Reference

1. Refer to “Filter Categories” on page 2-1.
2. For information on lowpass filter behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
3. This filter is unidirectional. Its input impedance is infinite and its output impedance is specified by \( Z_{out} \).
4. This Gaussian filter is used in the GMSK modulation system.
5. In the steady-state frequency domain analysis, an ideal frequency-domain response is used. However, this ideal response has an infinite duration impulse response that must be approximated for time domain simulations in either Transient or Circuit Envelope. If DelayBits is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.

6. A value of 1.0 will be used internally when DelayBits is set to a value that is less than 1.0.

7. Accuracy of this model in Transient or Circuit Envelope can be further controlled through the ImpMaxFreq, ImpDeltaFreq, and ImpMaxPts parameters.

8. This component has no default artwork associated with it.
LPF_PoleZero (Lowpass Filter, Pole Zero)

Symbol

Parameters
Poles = list of poles
Zeros = list of zeros
Gain = gain factor
Fpass = 3dB bandwidth
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1
Z2 = reference impedance for port 2

Notes/Equations
1. This is an S-domain filter.
2. Poles and Zeros are a list of complex pole/zero locations.
   The transfer function for the filter is:
   \[ S_{21} = \frac{(S - \text{Zero1})(S - \text{Zero2}) \ldots}{(S - \text{Pole1})(S - \text{Pole2}) \ldots} \]
   where
   \[ S = \frac{\text{Freq}}{\text{Fpass}} \]
   and
   Freq is the analysis frequency
   At least one pole must be supplied.
3. For information on lowpass filter behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
LPF_Polynomial (Lowpass Filter, Polynomial)

Symbol

Parameters
Numerator = list of numerator coefficients
Denominator = list of denominator coefficients
Gain = gain factor
Fpass = 3dB bandwidth
StopType = stopband input impedance type: OPEN or SHORT
Z1 = reference impedance for port 1, in ohms
Z2 = reference impedance for port 2, in ohms

Notes/Equations
1. This is an S-domain filter.
2. Denominator and Numerator are a list of polynomial coefficients. The transfer function for the filter is:

\[
S_{21} = \text{Gain} \left( \frac{N_0 + (N_1 \times S) + (N_2 \times S^2)\ldots}{D_0 + (D_1 \times S) + (D_2 \times S^2)\ldots} \right)
\]

where

\[ S = j(Freq / Fpass) \]

and

Freq is the analysis frequency

At least one denominator coefficient must be supplied.

3. For information on lowpass filter behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.
LPF_RaisedCos (Lowpass Filter, Raised-Cosine)

Symbol

Parameters

Alpha = Rolloff factor defining filter excess bandwidth. Default = 0.35
SymbolRate = Digital symbol rate defining filter bandwidth, in Hz. Default = 24.3 KHz
DelaySymbols = Number of symbols delayed by filter. Default = 5
Exponent = Exponent factor, to provide for Root Raised-Cosine filter. Default = 0.5
DutyCycle = Pulse duty cycle in percent, used for sinc(x) correction. Default = 0
SincE = Flag to include the Exponent factor on the sinc(x) correction (yes/no). Default = no
Gain = Gain normalization factor. Default = 1.0
Zout = Output impedance, in Ohms. Default = 50 Ohms
WindowType = Window type applied to impulse response. 0=None (default). 1=Hann, 2=Hamming
ImpMaxFreq = Maximum frequency to consider when calculating impulse response, in Hz.
ImpDeltaFreq = Frequency sample spacing when calculating impulse response, in Hz.
ImpMaxPts = Maximum number of points in impulse response. Default = 4096
Other = output string to netlist

Range of Usage

0 ≤ Alpha ≤ 1
DelaySymbols ≥ 1
0 ≤ Exponent ≤ 1
0 ≤ DutyCycle ≤ 100
Notes/Equations

1. Refer to “Filter Categories” on page 2-1.

2. For information on lowpass filter behavior at DC, refer to “Lowpass Filter Behavior at DC” on page 2-4.

3. This filter is unidirectional; input impedance is infinite; output impedance is specified by Zout.

4. The voltage gain is described by the following function. 

\[
\frac{V_{out}}{V_{in}} = \frac{Gain \times G_{filt} \times e^{-j2\pi frequency(DelaySymbols/\text{SymbolRate})}}{G_{comp}}
\]

where 

\(G_{filt} = \begin{cases} 
1 & \text{for frequency } < 0.5 \times (1-\text{Alpha}) \times \text{SymbolRate} \\
0 & \text{for frequency } > 0.5 \times (1+\text{Alpha}) \times \text{SymbolRate} \\
0.5(1-\sin(\pi \times (\text{frequency} - 0.5 \times \text{SymbolRate})/\text{SymbolRate}))^{\text{Exponent}} & \text{otherwise}
\end{cases} \)

\(G_{comp} = \begin{cases} 
1.0 & \text{if DutyCycle } = 0, \text{ else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)]^{\text{Exponent}} & \text{if SincE } = \text{YES, else} \\
[0.01 \times \text{DutyCycle} \times \text{sinc}(x)] & \text{if SincE } = \text{NO}
\end{cases} \)

Freq = abs (Fcenter - frequency)

\(\text{sinc}(x) = \sin(x)/x\)

\(x = 0.01 \times \text{DutyCycle} \times \pi \times \text{frequency}/\text{SymbolRate}\)

5. While the Exponent can be any value, the two standard values are 1.0 for the ideal raised-cosine filter response or 0.5 to simulate the root raised-cosine filter response when present at both the receiving and transmitting channels.

6. In the steady-state, frequency-domain analyses, the ideal frequency-domain response described previously is used. However, this ideal response has an infinite duration impulse response which must be approximated for time domain simulations in either transient or circuit envelope. If DelaySymbols parameter is set too small, then the impulse response will be severely truncated and will not accurately reflect the ideal frequency response.
A DelaySymbols parameter of 15 should result in saturated frequency domain sidelobes of -75dBc or smaller. This number is approximate and represents the saturated sidelobe level at frequencies far greater than the filter’s cutoff frequency. The sidelobes at, say, twice the filter’s cutoff frequency have generally not saturated and will typically be higher than -75 dBc. The saturated sidelobe level may depend on whether a transient or circuit envelope simulation is performed and on the window type used. It is significantly lower than -75dBc in many cases. The accuracy of this model in transient or circuit envelope simulations can be further controlled through the ImpMaxFreq, ImpDeltaFreq and ImpMaxPts parameters.

7. The filter can include gain equalization to compensate for duty cycle roll-off. If DutyCycle = 0.0, then no compensation will be applied. If SincE= YES, Exponent will be applied to the gain compensation term, $G_{comp}$. (Note that the Exponent term is always present in the $G_{filt}$ term.)

8. This component has no default artwork associated with it.
SAW Filter (Saw Filter)

Symbol

Parameters

- $F_{\text{center}}$: passband center frequency
- $IL$: passband insertion loss, in dB
- $BW_{\text{pass}}$: passband edge-to-edge width
- $A_{\text{pass}}$: attenuation at passband edges, in dB
- $BW_{\text{stop}}$: stopband edge-to-edge width
- $A_{\text{stop}}$: attenuation at stopband edges, in dB
- $G_{\text{Delay}}$: group delay
- $A_{\text{ripple}}$: passband amplitude ripple, in dB
- $\Phi_{\text{Ripple}}$: passband phase ripple from linear phase, in degrees
- $\text{MaxRej}$: maximum rejection level, in dB
- $Z_{\text{in}}$: input impedance
- $Z_{\text{out}}$: output impedance
- $\text{WindowType}$: Window Type, 0 = None, 1 = Hann, 2 = Hamming
- $\text{Temp}$: temperature, in °C
- $\text{Other}$: output string to netlist

Range of Usage

- $BW_{\text{stop}} \geq BW_{\text{pass}}$
- $G_{\text{Delay}} \geq \pi/BW_{\text{stop}}$

Notes/Equations

1. SAW Filter models the typical behavior of SAW bandpass filters. If the $\Phi_{\text{Ripple}}$ parameter is not specified, the filter will have perfect linear phase.
2. To maintain causality, $G_{\text{Delay}}$ must be set to at least $\pi/BW_{\text{stop}}$. 
Chapter 3: Modulators and Demodulators

Introduction

The Filters - <filter type> and System - <device type> palettes contain two fundamentally different types of behavioral system models.

Filters, System - Amps & Mixers, and System - Mod/ Demod can be classified as tops-down system models that support a tops-down system design flow where model behaviors are characterized by a small number of independent parameters such as frequency, power and load. They are often referred to as parameter-based behavioral models.

System - Data Models can be classified as bottoms-up system models that support a bottoms-up verification flow where model behaviors are extracted from a simulation (or measurement) of a transistor-level circuit. They are often referred to as data-based behavioral models.

The parameter-based behavioral models typically provide superior speed relative to the data-based behavioral models with both of these being vastly superior to a brute-force transistor-level simulation.

The data-based behavioral models typically provide superior accuracy relative to the parameter-based behavioral models as they capture actual behaviors of implemented circuit components and not just design specifications.

The differences between parameter- and data-based behavioral models justify a palette emphasis on flow (all data-based behavioral models grouped together) rather than functionality (all amplifiers, mixers, modulators, and demodulators grouped together) and resulted in the addition of a System - Data Models palette.

The use model for parameter-based behavioral models is to simply set a series of parameters prior to using the model. The use model for data-based behavioral models is slightly more involved. For a discussion, refer to Chapter 8, System Data Models.

The modulators and demodulators system model library contains time domain tuned modulators and tuned demodulators. Each component in this library is described following this introduction.

When using the System-Mod/Demod components in the Analog/RF schematic, it is important to note that terminating the component with a load resistance equivalent to the component output resistance will provide an output voltage that is half of the
applied input voltage. Consider the Thevenin equivalent of the output of a Mod/Demod component.

\[ V_{out} = \frac{V_o \times \text{Load Resistance}}{\text{Load Resistance} + \text{Output Resistance}} \]

In an Analog/RF schematic, the value for \( V_{out} \) will be \( \frac{1}{2} \) \( V_o \) when Output Resistance=Load Resistance. In general, \( V_{out} = \frac{V_o \times \text{Load Resistance}}{\text{Load Resistance} + \text{Output Resistance}} \). Thus, this is the potential divider action. All of the components in the System-Mod/Demod library have this property except for N_StateDemod and PM_UnwrapDemodTuned. (These two components do not have an \( R_{out} \) parameter).

For a similar circuit in a DSP schematic, the effect of the potential divider on the output voltage will not be noticeable when the output resistance equals the load resistance. In this case, \( V_{out} \) equals \( V_o \). For the DSP components, there is an additional factor of 2 at the output voltage to cancel the factor of \( \frac{1}{2} \) from the potential divider. For a description of the digital implementation of the modulators and demodulators, and the potential divider action, refer to Introduction: Timed, Modem Components in the Signal Processing component documentation.
AM_DemodTuned (AM Demodulator, Tuned)

Symbol

Parameters

Fnom = nominal input frequency, in hertz
Rout = output resistance, in ohms

Notes/Equations

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous amplitude of the selected carrier frequency.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

   \[ V_{out_0} = \text{mag}(Vin_k) \]

   where

   \[ \text{mag}(V) = \sqrt{(Re(V))^2 + (Im(V))^2} \]

   and

   \( k \) represents the value at the analysis harmonic frequency closest to the Fnom value

   \( 0 \) represents the baseband component of the output voltage. All non-baseband output frequency components are 0.

2. This model generates only the absolute value of the input because the imaginary part of baseband signals is 0 for transient and baseband envelope simulations.
**AM_ModTuned (AM Modulator, Tuned)**

**Symbol**

![Symbol Image]

**Parameters**

- **ModIndex** = modulation index
- **Fnom** = nominal input frequency, in hertz
- **Rout** = output resistance, in ohms

**Notes/Equations**

1. This is a tuned modulator that selects the input harmonic defined by the specified **Fnom** frequency and amplitude modulates it by the baseband modulation input.

   If there is no analysis harmonic frequency close enough to the **Fnom** frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by **Rout**, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

   \[ V_{out_k} = (1 + \text{ModIndex} \times V_{mod_0}) \times V_{in_k} \]

   where

   - \( k \) represents the value at the analysis harmonic frequency closest to the **Fnom** value
   - \( 0 \) represents the baseband component of the modulation input voltage.

2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting AM signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.

3. This model works in transient and baseband envelope simulations. Because the describing equation is valid for baseband signals, **Fnom** has no effect.
**FM_DemodTuned (FM Demodulator, Tuned)**

**Symbol**

```
\[ <\text{symbol}> \]
```

**Parameters**

- **Sensitivity**: demodulation sensitivity, in hertz/volt
- **Fnom**: nominal input frequency, in hertz
- **Rout**: output resistance, in ohms

**Notes/Equations**

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous frequency of the selected carrier frequency.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm.

   The instantaneous frequency difference is equal to the time derivative of the instantaneous phase. This is approximated using a simple finite difference. As long as the phase does not change by greater than \( 180^\circ \) in one time step, the frequency calculation is non-ambiguous. The open circuit output voltage can be described as:

   \[
   V_{out_0} = \frac{\Phi(V_{in_k}(t)) - \Phi(V_{in_k}(t - \text{timestep}))}{2\pi \times \text{Sensitivity} \times \text{timestep}}
   \]

   where

   \[
   \Phi(V) = \frac{\text{Im}(V)}{\text{Re}(V)}
   \]

   \( k \) represents the value at the analysis harmonic frequency closest to the Fnom value

   \( 0 \) represents the baseband component of the output voltage

   All non-baseband output frequency components are 0.

2. For transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.
Modulators and Demodulators

**FM ModTuned (FM Modulator, Tuned)**

**Symbol**

```
\[ \text{Symbol} \]
```

**Parameters**

- Sensitivity = modulation sensitivity, in hertz/volt
- Fnom = nominal input frequency, in hertz
- Rout = output resistance, in ohms

**Notes/Equations**

1. This is a tuned modulator that selects the input harmonic defined by the specified Fnom frequency and frequency modulates it by the baseband modulation input.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

   \[
   V_{out_k} = e^{j2\pi \text{Sensitivity} \int_0^t V_{m_k} dt} \times V_{in_k}
   \]

   where

   - \( k \) represents the value at the analysis harmonic frequency closest to the Fnom value
   - 0 represents the baseband component of the modulation input voltage.

   For time \( t=0 \), the integrator of the modulation input signal is reset to 0. In all steady-state analyses such as harmonic balance, dc, and ac, time is kept at 0, so no modulation occurs.

2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting FM signal is present at pin 2. The RF carrier should be a
frequency-domain source; the modulating signal should be a time-domain source.

3. This model will not function as a frequency modulator in transient and baseband envelope simulations because of the use of the complex exponential operator.
Modulators and Demodulators

**IQ_DemodTuned (I/Q Demodulator, Tuned)**

**Symbol**

![Symbol Image]

**Parameters**

- $F_{nom} =$ nominal input frequency, in hertz
- $R_{out} =$ output resistance, in ohms

**Notes/Equations**

1. This is a tuned demodulator that selects the input harmonic closest to the specified $F_{nom}$ frequency and generates two baseband output signals equal to the instantaneous in-phase and quadrature-phase components of the selected carrier frequency.

   If there is no analysis harmonic frequency close enough to the $F_{nom}$ frequency, a warning is issued and the output is 0. Input impedance is infinite; output impedance is set by $R_{out}$, and is limited to a minimum value of 0.1 ohm. The open circuit output voltages are defined simply by

   $$V_{i_0} = \Re(V_{in_k})$$
   $$V_{q_0} = \Im(V_{in_k})$$

   where

   - $k$ represents the value at the analysis harmonic frequency closest to the $F_{nom}$ value
   - 0 represents the baseband component of the output voltages.
   - All non-baseband output frequency components are 0.

2. In transient and baseband envelope simulations, because the imaginary part of baseband signals is 0, this model replicates the input on the $i$ output only.
**IQ_ModTuned (I/Q Modulator, Tuned)**

**Symbol**

![Symbol Diagram]

**Parameters**

- F<sub>nom</sub> = nominal input frequency, in hertz
- Rout = output resistance, in ohms

**Notes/Equations**

1. This is a tuned modulator that selects the input harmonic defined by the specified F<sub>nom</sub> frequency and modulates it according to the I (in-phase) and Q (quadrature) modulation inputs.
   
   If there is no analysis harmonic frequency close enough to the F<sub>nom</sub> frequency, a warning is issued and the output is 0. All three input impedances are infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by
   
   \[ V_{out_k} = (V_{10} + j \times V_{Q0}) \times V_{in_k} \]
   
   where
   - \( k \) represents the value at the analysis harmonic frequency closest to the F<sub>nom</sub> value
   - \( 0 \) represents the baseband component of the two modulation input voltages.

2. The RF carrier is injected at pin 1; I data is injected at pin 3; Q data is injected at pin 4. The resulting modulated signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signals should be a time-domain source.

3. This model will not function as a phase modulator in transient and baseband envelope simulation because of the use of the complex j operator.

4. Both AM and PM modulation can be generated.
Modulators and Demodulators

**N_StateDemod (N-State Demodulator)**

**Symbol**

![Symbol Image]

**Parameters**

Fnom = nominal input frequency, in hertz

StateArray = complex array of state values

**Notes/Equations**

1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output state signal representing the nearest complex nominal state of the selected carrier at the time of last rising clock edge.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. On each rising clock edge, the I/Q information of the carrier is sampled. Using a Euclidean distance measure, the closest nominal state in the StateArray variable is determined and that state number, minus 1, is output as the baseband state voltage. This can be considered a 2-dimensional quantizer.

2. The nominal state locations are arbitrary and user definable in the StateArray variable, which can be created from a list of complex values using the list() function.

3. In transient and baseband envelope simulations, this state demodulation model functions correctly and could be used as a 1-dimensional arbitrary state quantizer.
N_StateMod (N-State Modulator)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxStates</td>
<td>maximum number of input states</td>
</tr>
<tr>
<td>StateArray</td>
<td>complex array of state values</td>
</tr>
<tr>
<td>Fnom</td>
<td>nominal input frequency, in hertz</td>
</tr>
<tr>
<td>Rout</td>
<td>output resistance, in ohms</td>
</tr>
</tbody>
</table>

Notes/Equations

1. This N-state modulator can be used to create an arbitrary, user-defined constellation of complex modulation states. The input harmonic closest to the specified Fnom frequency is selected and modulated by a sequence of complex states.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The states are defined by the complex array variable StateArray. The sequence of states is determined by the baseband component of the state modulation input. The two input impedances are infinite; output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

   \[
   V_{out_k} = \text{State}(\text{int}(V_{s_0}) + 1) \times V_{in_k}
   \]

   where

   \( k \) represents the value at the analysis harmonic frequency closest to the Fnom value
   
   \( 0 \) represents the baseband component of the modulation state input voltage; this input voltage is internally limited to be between 0 and MaxStates -1.

   The StateArray variable can be created from list of complex values using the list() function.
2. This model will not function as a complex modulator in transient and baseband envelope simulations if complex state values are specified.

3. **Figure 3-1** and **Figure 3-2** show a schematic example and simulation results for a 16-state modulator being swept linearly through all 16 of its states.

**Figure 3-1. 16-State Modulation Example**

**Figure 3-2. Simulation Results**
PI4DQPSK_ModTuned (PI/4 DQPSK Modulator, Tuned)

Symbol

Parameters

- $F_{nom} =$ nominal input frequency, in hertz
- $R_{out} =$ output resistance, in ohms
- $SymbolRate =$ output symbol rate (one-half input bit rate)
- $Delay =$ sampling delay, in seconds

Notes/Equations

1. This tuned PI/4 DQPSK modulator selects the input harmonic closest to the specified $F_{nom}$ frequency and modulates it according to the phase state determined by differentially encoding the input bit stream and applying the PI/4 phase offset.

   If there is no analysis harmonic frequency close enough to the $F_{nom}$ frequency, a warning is issued and the output is 0. The input bit stream is sampled at a rate determined by the $SymbolRate$ with an initial synchronizing delay determined by the $Delay$ parameter. The actual sampling rate is rounded to an integer multiple of the system timestep—use a corresponding discrete time source to generate this serial bit stream or ensure that the analysis sampling rate is an integer multiple of the bit rate. Two consecutive bits are monitored, along with the previous phase state, to determine the next phase state. An input bit is assumed to be 1 if it is greater than 0.5V and 0 otherwise.

2. The RF carrier is injected at pin 1; the input bit stream is injected at pin 3. The resulting PI4DQPSK signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.

3. The input impedances of both the serial bit stream input and the RF carrier are infinite; output impedance is set by $R_{out}$, and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by:
where

$k$ represents the value at the analysis harmonic frequency closest to the $F_{nom}$ value

0 represents the baseband component of the bit stream input voltage.

<table>
<thead>
<tr>
<th>$V_m_i(t)$</th>
<th>$V_{m_i}(t-\text{BitTime})$</th>
<th>Phase Transition = $\text{Next Phase} - \text{Current Phase}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>45°</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>135°</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-45°</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-135°</td>
</tr>
</tbody>
</table>

4. This model will not function as a modulator in transient and baseband envelope simulations because of the use of complex exponential functions.
PM_DemodTuned (PM Demodulator, Tuned)

Symbol

Parameters
Sensitivity = demodulation sensitivity, in degree/volt
Fnom = nominal input frequency, in hertz
Rout = output resistance, in ohms

Notes/Equations
1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous phase of the selected carrier frequency.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The input impedance is infinite. The output impedance is set by Rout, and is limited to a minimum value of 0.1 ohm.

   The open circuit output voltage is defined by

   \[ V_{out_0} = \frac{\Phi(V_{in_k}(t)) \times 180}{\pi \times \text{Sensitivity}} \]

   where

   \[ \Phi(V) = \tan^{-1}\left(\frac{\text{Im}V}{\text{Re}V}\right) \]

   \( k \) represents the value at the analysis harmonic frequency closest to the Fnom value.

   \( 0 \) represents the baseband component of the output voltage.

   All non-baseband output frequency components are 0.

   The maximum phase range of this demodulator is ±180°.

2. In transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.
PM_ModTuned (PM Modulator, Tuned)

Symbol

Parameters

- Sensitivity = demodulation sensitivity, in degree/volt
- \( F_{\text{nom}} \) = nominal input frequency, in hertz
- \( R_{\text{out}} \) = output resistance, in ohms

Notes/Equations

1. This is a tuned modulator that selects the input harmonic defined by the specified \( F_{\text{nom}} \) frequency and phase modulates it by the baseband modulation input. If there is no analysis harmonic frequency close enough to the \( F_{\text{nom}} \) frequency, a warning is issued and the output is 0. The two input impedances are infinite. The output impedance is set by \( R_{\text{out}} \), and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by

\[
V_{\text{out}} = e^{j\pi S_{\text{sensitivity}} \times (V_{\text{mod}})/180} \times V_{\text{in}}
\]

where

- \( k \) represents the value at the analysis harmonic frequency closest to the \( F_{\text{nom}} \) value
- \( 0 \) represents the baseband component of the modulation input voltage.

2. The RF carrier is injected at pin 1; the modulating signal is injected at pin 3. The resulting PM signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.

3. This model will not function as a phase modulator in transient and baseband envelope simulations because of the use of the complex exponential operator.
PM_UnwrapDemodTuned (PM Unwrapped Demodulator, Tuned)

Symbol

Parameters
Sensitivity = modulation sensitivity, in degree/volt
Fnom = nominal input frequency, in hertz
MaxAngle = unwrapped phase angle range, in degrees

Notes/Equations
1. This is a tuned demodulator that selects the input harmonic closest to the specified Fnom frequency and generates a baseband output signal equal to the instantaneous unwrapped phase of the selected carrier frequency.
   If there is no analysis harmonic frequency close enough to the Fnom frequency, a warning is issued and the output is 0. The input impedance is infinite. The output impedance is 0.
2. The phase range of this demodulator is equal to ± MaxAngle. The phase at time 0 is set equal to the normal 180° phase of the selected carrier frequency. By tracking modulo 360° phase transitions, the unwrapped phase, relative to the initial time 0 value, is calculated. The user can then set the desired modulo range with the MaxAngle parameter. A large number will generate totally unwrapped phase.
   This unwrapping of phase works for time sweeps in envelope analyses; it cannot be used to unwrap the phase of frequency or other parameter sweeps.
3. In transient and baseband envelope simulations, this model does not generate the expected results because the imaginary part of baseband signals is 0.
QPSK_ModTuned (QPSK Modulator, Tuned)

Symbol

Parameters

F_{nom} = nominal input frequency, in hertz
R_{out} = output resistance, in ohms
State1 = complex modulation coordinates of State 1
State2 = complex modulation coordinates of State 2
State3 = complex modulation coordinates of State 3
State4 = complex modulation coordinates of State 4

Notes/Equations

1. This tuned quadrature phase shift keying (QPSK) modulator is actually an arbitrary 4-state modulator. The input harmonic closest to the specified F_{nom} frequency is selected and modulated by one of four user-defined complex values. If there is no analysis harmonic frequency close enough to the F_{nom} frequency, a warning is issued and the output is 0.

The complex values are determined by the baseband component of the state modulation input. An input less than 0.5V selects State 1; less than 1.5V selects State 2; less than 2.5V selects State 3; and, any input greater than 2.5V selects State 4. The two input impedances are infinite. The output impedance is set by Rout and is limited to a minimum value of 0.1 ohm. The open circuit output voltage is defined by:

\[ V_{out_k} = \text{State}[\text{int}(V_{s0} + 1.5)] \times V_{in_k} \]

where

\( k \) represents the value at the analysis harmonic frequency closest to the F_{nom} value
\( 0 \) represents the baseband component of the modulation state input voltage.
2. The RF carrier is injected at pin 1; the input bit stream is injected at pin 3. The resulting QPSK signal is present at pin 2. The RF carrier should be a frequency-domain source; the modulating signal should be a time-domain source.

3. This model will not function as a complex modulator in transient and baseband envelope simulations if the specified state values are complex.
Modulators and Demodulators

3-20  QPSK_ModTuned (QPSK Modulator, Tuned)
Chapter 4: Passive System Components
AntLoad (Antenna Load)

Symbol

Illustration

Parameters
AntType = antenna type: MONOPOLE or DIPOLE
Length = physical antenna length, in specified units
RatioLR = length-to-radius ratio

Range of Usage
monopole simulation frequency (MHZ) ≤ \frac{287}{L(\text{meters})} \times \frac{\text{RatioLR}}{\text{RatioLR} + 1}
dipole simulation frequency (MHZ) ≤ \frac{575}{L(\text{meters})} \times \frac{\text{RatioLR}}{\text{RatioLR} + 2}

Notes/Equations
1. This component models the input impedance of a monopole or dipole antenna. Transmission is not modeled.
2. For time-domain analysis, the frequency-domain analytical model is used.
3. This component has no default artwork associated with it.
References


Attenuator (Attenuator)

Symbol

Parameters
Loss = attenuation, in dB
VSWR = voltage standing wave ratio for both ports
Rref = reference resistance for both ports
Temp = temperature, in °C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
ReturnPhase = phase scaling of S_{11}, S_{22}; default is 0

Range of Usage
VSWR > 1.0

Notes/Equations
1. The S-parameters for Attenuator are:
   \[ |S_{12}| = |S_{21}| = 10^{-\frac{|\text{Loss}|}{20}} \]
   \[ \text{Phase}(S_{12}) = \text{Phase}(S_{21}) = 0 \]
   \[ |S_{11}| = |S_{22}| = \frac{(\text{VSWR} - 1)}{(\text{VSWR} + 1)} \] (regarding limiting, refer to Note 2)
   \[ \text{Phase}(S_{11}) = \text{Phase}(S_{22}) = -90 \text{ if ReturnPhase} < 0 \]
   \[ \text{Phase}(S_{11}) = \text{Phase}(S_{22}) = 0 \text{ if ReturnPhase} = 0 \]
   \[ \text{Phase}(S_{11}) = \text{Phase}(S_{22}) = 90 \text{ if ReturnPhase} > 0 \]

2. When ReturnPhase=0, |S_{11}| and |S_{22}| will be limited to \( \min(|S_{11}|, 1 - |S_{21}|) \) in order to maintain passivity constraints.

3. This component will always provide attenuation; for example, either Loss=20 or Loss=-20 will result in 20 dB attenuation.
4. When used in time domain simulations, set ReturnPhase=0; any other
ReturnPhase value will not produce proper results because (+/-)j in the
frequency domain corresponds to a non-causal impulse response in time
domain.

5. When checking passivity, remember that infinitesimal numerical differences
can make a difference. For example, an Attenuator with

\[
|S_{21}| = |S_{12}| = 0.8
\]
\[
|S_{11}| = |S_{22}| = \sqrt{1-|S_{11}|^2} = 0.6
\]

ReturnPhase <> 0

is theoretically passive. However, it may give a non-passivity warning in ADS
due to infinitesimal numerical differences. Subtracting a small number, say
1e-12, from all S-parameters should address the problem without causing
noticeable changes in the results.

6. CheckPassivity controls whether a separate passivity check is enabled. This is
what outputs the Linear S_Port ... is non-passive warning. ReturnPhase
controls whether limiting is enabled. This is what outputs the VSWR limited ...
warning. When limiting is enabled, it implies a passivity check to find out what
limiting, if any, should be done (refer to note 7). This passivity check is silent
and never produces a non-passivity warning. When ReturnPhase=0, VSWR
limiting is done before the passivity check if CheckPassivity=yes. This means
that for ReturnPhase=0, a non-passivity warning will never be issued
regardless of whether CheckPassivity is set to yes or no. For CheckPassivity=no,
a passivity check will not be done; therefore a non-passivity warning will not be
output. For CheckPassivity=yes, a passivity check will be done but the
parameters have already been limited to ensure passivity so the component will
be found to be passive and a non-passivity warning will not be output.

7. This note answers two questions regarding Attenuator:

Q How does Attenuator do its passivity check and how/why does it perform
limiting to ensure passivity?

A The S-matrix for this attenuator is

\[
S = \begin{bmatrix}
S_{11} & S_{12} \\
S_{21} & S_{22}
\end{bmatrix}
\]

We have
Passive System Components

\[
S = \frac{-\text{abs}(\text{Loss})}{10^{20}} = T
\]

and we see that \( T \) is real and fulfills \( 0 \leq T \leq 1 \) for Loss values from minus infinity to infinity. We also have

\[
|S_{11}| = |S_{22}| = \text{scale} \times \frac{\text{VSWR}_2 - 1}{\text{VSWR}_2 + 1} = R
\]

where

\[
\text{scale} = \begin{cases} 
1 & \text{if (ReturnPhase=0)} \\
-j & \text{if (ReturnPhase<0)} \\
\end{cases}
\]

\[
\text{VSWR}_2 = \begin{cases} 
\min(\text{VSWR}, \text{MaxVSWR}) & \text{if (ReturnPhase=0)} \\
\text{VSWR} & \text{else} \\
\end{cases}
\]

\[
\text{MaxVSWR} = \frac{(1+\text{MaxR})}{(1-\text{MaxR})}
\]

\[
\text{MaxR} = \text{maximum value of R which guarantees passivity, TBD}
\]

and we see that \( R \) is real for \( \text{ReturnPhase}=0 \) and fulfills \( 0 \leq R \leq 1 \) for VSWR values from 1 to infinity. The S-matrix for \( \text{ReturnPhase}=0 \) is therefore

\[
S = \begin{bmatrix} R & T \\ T & R \end{bmatrix}
\]

The condition for passivity is that eigenvalues((transpose(conjugate(S)) \times S) \leq 1. For the real and symmetric matrix above, this reduces to eigenvalues(S \times S) \leq 1.

We have

\[
S \times S = \begin{bmatrix} R^2 + T^2 & 2 \times R \times T \\ 2 \times R \times T & R^2 + T^2 \end{bmatrix}
\]

and find the eigenvalues \( R^2-2 \times R \times T+T^2 \) and \( R^2+2 \times R \times T+T^2 \). Letting these eigenvalues equal to 1 (the max value for passivity), we get

\[
R^2-2 \times R \times T+T^2=1 \rightarrow R=1+T \quad \text{and} \quad R=1-T
\]

\[
R^2+2 \times R \times T+T^2=1 \rightarrow R=1-T \quad \text{and} \quad R=1-T
\]

and see that eigenvalues \( (S \times S) \leq 1 \) implies \( R \geq -1+T, R \leq 1+T, R \geq -1-T \) and \( R \leq 1-T \). For \( 0 \leq R, T \leq 1 \), the first three inequalities are always satisfied and the last one gives MaxR=1-T.
Q I have a dissipationless attenuator. It has a given S21 and because it is dissipationless I know that
\[ |S_{11}| = \sqrt{1 - |S_{21}|^2}. \]
For instance, I might have S21=0.8 and I know that |S11| =0.6 corresponding to a VSWR of 4. I enter S21=0.8 and S11=0.6 for the attenuator but when I simulate I’m told that VSWR is limited to 1.5 corresponding to an S11 value of 0.2. I know that my attenuator is passive with |S11| =0.6. Isn’t the limiting by Attenuator too restrictive and wrong?
A Let’s examine the above S-matrix. It is
\[
\begin{bmatrix}
0.6 & 0.8 \\
0.8 & 0.6
\end{bmatrix}
\]
and it does fulfill
\[ \sum_{i=1}^{2} (|S_{ij}|^2) = 1 \quad [j = 1, 2] \]
However, for a dissipationless network it must also fulfill
\[ \sum_{i=1}^{2} (S_{ij} \times \text{conjugate}(S_{iq})) = 0 \quad [j = 1, 2; \ q = 1, 2; \ q < > j] \]
and it clearly does not. What is wrong is that S11=S22 have zero phase.
To remedy this, we could consider
\[
\begin{bmatrix}
j \times 0.6 & 0.8 \\
0.8 & j \times 0.6
\end{bmatrix}
\]
which fulfills both of the above equations. A similar matrix could be constructed by scaling with -j. These matrices must be entered into ADS with ReturnPhase=0 and ReturnPhase<0, respectively, not the default ReturnPhase=0. For ReturnPhase<>0, no limiting is done on VSWR and the above concern is void. An Attenuator with the original S-matrix is not dissipationless and must be limited in order to be passive.
All this generalizes to $N \times N$ matrices. Look, for example, at a standard four-port directional coupler. The elements are not all real. There are $j$'s floating around. Why? This is why.

A quick check shows that the eigenvalues of

$$
\begin{pmatrix}
0.2 & 0.8 \\
0.8 & 0.2 \\
\end{pmatrix}
\begin{pmatrix}
0.2 & 0.8 \\
0.8 & 0.2 \\
\end{pmatrix}
$$

are 0.36 and 1.0 and that the eigenvalues of

$$
\begin{pmatrix}
0.6 & 0.8 \\
0.8 & 0.6 \\
\end{pmatrix}
\begin{pmatrix}
0.6 & 0.8 \\
0.8 & 0.6 \\
\end{pmatrix}
$$

are 0.04 and 1.96 showing again that the circuit corresponding to the former/limited matrix is passive (largest eigenvalue of $S \times S$ equals one) while the circuit corresponding to the latter/unlimited matrix is not passive (largest eigenvalue of $S \times S > 1$).
Balun3Port (Balun, 3-port)

Symbol

Parameters
None

Notes/Equations

1. Balun3Port realizes the ideal transformation between a balanced differential-mode signal and unbalanced, single-ended signals. It can be useful to connect a source to a differentially fed circuit, although it does ignore common-mode effects.

2. Balun3Port realizes the voltage and current transformations given by:

   \[ v_d = v_+ - v_- \]
   \[ i_+ = -i_- = -i_d \]

   where

   \[ v_d/i_d = \text{the differential mode voltage/current at pin } d \]
   \[ v_4/i_+ = \text{the single line voltage/current at pin } + \]
   \[ v_-/i_- = \text{the single line voltage/current at pin } - \]


   The minus signs in the current definitions are due to the standard definition of currents directed into the Balun3Port component.

3. An equivalent functionality can be realized with a Balun4Port that has the common-mode pin grounded. However, the Balun3Port provides better convergence properties.

4. Balun3Port is bi-directional. When fed at the differential-mode pin, it realizes the transformations:

   \[ v_+ = -v_- = (v_d)/2 \]
   \[ i_+ = -i_- = -i_d \]
Passive System Components

5. Examples of using Balun3Port to convert between (unbalanced) ADS sources and balanced circuits can be found in the ADS examples directory; access these examples from the ADS Main window > File > Example Project.

- **RFIC > MixerDiffMode_prj** demonstrates the use of Balun4Port to present differential-mode sources (as well as common mode biases) to the RF and LO inputs. It also shows the use of Balun3Port to single-ended (differential-mode) IF output, which is needed to properly calculate the noise figure.

- **BehavioralModels > DifferentialModels_prj** demonstrates the use of Balun3Port and Balun4Port in conjunction with single-ended System-Data Models in order to create a data-based behavioral model of a differentially fed mixer.
Balun4Port (Balun, 4-port)

Symbol

Parameters
None

Notes/Equations
1. Balun4Port realizes the ideal transformation between balanced (differential- and common-mode) signals and unbalanced (single-ended) signals. It can be used to connect sources to a differentially fed circuit, particularly when modeling common-mode effects are important.

2. Balun4Port realizes voltage and current transformations given by:

- \( v_d = v_+ - v_- \)
- \( v_c = (v_+ + v_-)/2 \)
- \( i_d = -(i_+ - i_-)/2 \)
- \( i_c = -(i_+ + i_-) \)

where

- \( v_d/i_d = \) differential mode voltage/current at pin d
- \( v_c/i_c = \) common mode voltage/current at pin c
- \( v_+ / i_+ = \) single line voltage/current at pin +
- \( v_- / i_- = \) single line voltage/current at pin -


The minus signs in the current definitions are due to the standard definition of currents directed into the Balun4Port component.

3. Balun4Port is bi-directional. It converts common/differential-mode signals into two single-ended signals, as well as converting two single-ended signals into common/differential mode signals.
Passive System Components

4. If common-mode effects are not desired, Balun3Port provides an equivalent, but numerically more robust, result as grounding the common-mode pin of Balun4Port.

5. Examples of using Balun4Port to convert between (unbalanced) ADS sources and balanced circuits can be found in the ADS examples directory; access these examples from the ADS Main window > File > Example Project.

- RFIC > MixerDiffMode_prj demonstrates the use of Balun4Port to present differential-mode sources (as well as common mode biases) to the RF and LO inputs. It also shows the use of Balun3Port to single-ended (differential-mode) IF output, which is needed to properly calculate the noise figure.

- BehavioralModels > DifferentialModels_prj demonstrates the use of Balun3Port and Balun4Port in conjunction with single-ended System-Data Models in order to create a data-based behavioral model of a differentially fed mixer.
Balun6Port (Balun, 6-port)

Symbol

Parameters
None

Notes/Equations
1. Balun6Port is based on ideal transformers; it can be used to transform a single-ended signal to a differential signal with two ground planes such as a stripline.
2. This component passes DC.
## Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>first frequency breakpoint</td>
</tr>
<tr>
<td>F2</td>
<td>second frequency breakpoint</td>
</tr>
<tr>
<td>F3</td>
<td>third frequency breakpoint</td>
</tr>
<tr>
<td>Loss1</td>
<td>attenuation for frequencies ≤F1, in dB</td>
</tr>
<tr>
<td>Loss2</td>
<td>attenuation for frequencies &gt;F1 ≤F2, in dB</td>
</tr>
<tr>
<td>Loss3</td>
<td>attenuation for frequencies &gt;F2 ≤F3, in dB</td>
</tr>
<tr>
<td>Loss4</td>
<td>attenuation for frequencies &gt;F3, in dB</td>
</tr>
<tr>
<td>VSWR1</td>
<td>voltage standing wave ratio at both ports for frequencies ≤F1</td>
</tr>
<tr>
<td>VSWR2</td>
<td>voltage standing wave ratio at both ports for frequencies &gt;F1, ≤F2</td>
</tr>
<tr>
<td>VSWR3</td>
<td>voltage standing wave ratio at both ports for frequencies &gt;F2, ≤F3</td>
</tr>
<tr>
<td>VSWR4</td>
<td>voltage standing wave ratio at both ports for frequencies &gt;F3</td>
</tr>
<tr>
<td>Isolat</td>
<td>isolation, in dB</td>
</tr>
<tr>
<td>Z1</td>
<td>reference impedance for port 1</td>
</tr>
<tr>
<td>Z2</td>
<td>reference impedance for port 2</td>
</tr>
<tr>
<td>Z3</td>
<td>reference impedance for port 3</td>
</tr>
<tr>
<td>Temp</td>
<td>temperature, in degrees C</td>
</tr>
<tr>
<td>CheckPassivity</td>
<td>check passivity flag; if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.</td>
</tr>
</tbody>
</table>
Notes/Equations

1. Up to three frequency breakpoints can be used to define four bands. Each band will have a different Loss# and VSWR#. If no frequencies are specified, then this model is frequency independent and only uses Loss1 and VSWR1.

2. A loss is modeled regardless of the sign of the Loss parameter value.

3. All ports have the same VSWR.

4. Isolation is constant across all frequency bands.

5. The $3 \times 3$ S-matrix for Circulator is described by

\[
S_{11} = S_{22} = S_{33} = \frac{\text{VSWR} - 1}{\text{VSWR} + 1} \\
S_{12} = S_{23} = S_{31} = 10^{-\text{abs(Isolat)/20}} \\
S_{13} = S_{21} = S_{32} = 10^{-\text{abs(Loss)/20}}
\]
**Passive System Components**

**CouplerDual (Dual Coupler)**

**Symbol**

![Diagram of CouplerDual (Dual Coupler)]

**Parameters**

- **Coupling** = coupling factor, in dB
- **F1** = first frequency breakpoint
- **F2** = second frequency breakpoint
- **F3** = third frequency breakpoint
- **MVSWR1** = main arm VSWR for frequency ≤ F1
- **MVSWR2** = main arm VSWR for F1 < frequency ≤ F2
- **MVSWR3** = main arm VSWR for F2 < frequency ≤ F3
- **MVSWR4** = main arm VSWR for frequency > F3
- **CVSWR1** = coupled arm VSWR for frequency ≤ F1
- **CVSWR2** = coupled arm VSWR for F1 < frequency ≤ F2
- **CVSWR3** = coupled arm VSWR for F2 < frequency ≤ F3
- **CVSWR4** = coupled arm VSWR for frequency > F3
- **Loss1** = attenuation for frequency ≤ F1, in dB
- **Loss2** = attenuation for F1 < frequency ≤ F2, in dB
- **Loss3** = attenuation for F2 < frequency ≤ F3, in dB
- **Loss4** = attenuation for frequency > F3, in dB
- **Direct1** = directivity for frequency ≤ F1, in dB
- **Direct2** = directivity for F1 < frequency ≤ F2, in dB
- **Direct3** = directivity for F2 < frequency ≤ F3, in dB
- **Direct4** = directivity for frequency > F3, in dB
- **ZRef** = reference impedance for all ports
Temp = temperature, in degrees C

CheckPassivity = check passivity flag; if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

**Notes/Equations**

1. Port 1 is the input port; port 2 is the through port; port 3 is the coupled port; port 4 is the isolated port.

2. Both ports of the main arm (ports 1 and 2) are assumed to have the same VSWR. Both ports of the coupled arm (ports 3 and 4) are also assumed to have the same VSWR.

3. Coupling is assumed to be constant across all frequency bands.

4. Loss is the dissipation of the coupler; it affects S21 and S12 only. These S-parameters are calculated as if the coupler was dissipationless and are then scaled by

   \[ 10^{(-\text{abs}(\text{Loss})/20)} \]

5. Internally, coupler isolation is calculated as Coupling + Direct in dB.

6. Up to three frequency breakpoints can be used to define four bands, with different directivity losses and VSWR values for each frequency band. If no frequencies are specified, then this model is frequency independent and uses Direct1, Loss1, and VSWR1 only.

7. There is a 90-degree phase shift between the input and the coupled ports.

8. In the coupled arm, the two ports are uncoupled.
### CouplerSingle (Single Coupler)

**Symbol**

![CouplerDiagram](image)

**Parameters**

- **Coupling** = coupling factor, in dB
- **F1** = first frequency breakpoint
- **F2** = second frequency breakpoint
- **F3** = third frequency breakpoint
- **MVSWR1** = main arm VSWR for frequency $\leq F_1$
- **MVSWR2** = main arm VSWR for $F_1 < \text{frequency} \leq F_2$
- **MVSWR3** = main arm VSWR for $F_2 < \text{frequency} \leq F_3$
- **MVSWR4** = main arm VSWR for frequency $> F_3$
- **CVSWR1** = coupled arm VSWR for frequency $\leq F_1$
- **CVSWR2** = coupled arm VSWR for $F_1 < \text{frequency} \leq F_2$
- **CVSWR3** = coupled arm VSWR for $F_2 < \text{frequency} \leq F_3$
- **CVSWR4** = coupled arm VSWR for frequency $> F_3$
- **Loss1** = attenuation for frequency $\leq F_1$, in dB
- **Loss2** = attenuation for $F_1 < \text{frequency} \leq F_2$, in dB
- **Loss3** = attenuation for $F_2 < \text{frequency} \leq F_3$, in dB
- **Loss4** = attenuation for frequency $> F_3$, in dB
- **Direct1** = directivity for frequency $\leq F_1$, in dB
- **Direct2** = directivity for $F_1 < \text{frequency} \leq F_2$, in dB
- **Direct3** = directivity for $F_2 < \text{frequency} \leq F_3$, in dB
- **Direct4** = directivity for frequency $> F_3$, in dB
- **ZRef** = reference impedance for all ports
Temp = temperature, in degrees C

CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations

1. Port 1 is the input port; port 2 is the through port; port 3 is the coupled port.
2. Both ports of the main arm (ports 1 and 2) are assumed to have the same VSWR.
3. Coupling is assumed to be constant across all frequency bands.
4. Loss is the dissipation of the coupler; it affects S21 and S12 only. These S-parameters are calculated as if the coupler was dissipationless and are then scaled by
   \[ 10^{-\frac{\text{abs(Loss)}}{20}} \]
5. Internally, coupler isolation is calculated as Coupling + Direct in dB.
6. Up to three frequency breakpoints can be used to define four bands, with different directivity losses and VSWR values for each frequency band. If no frequencies are specified, then this model is frequency independent and uses Direct1, Loss1, and VSWR1 only.
7. There is a 90-degree phase shift between the input and the coupled ports.
8. In the coupled arm, the two ports are uncoupled.
Gyrator

Symbol

![Gyrator Symbol]

Illustration

![Gyrator Illustration]

Parameters

None

Notes/Equations

1. \( V_1 = \text{Ratio} \times I_2 \)
2. \( V_2 = -\text{Ratio} \times I_1 \)

\[
S_{11} = S_{22} = \frac{r^2 - 1}{r^2 + 1}
\]

\[
S_{21} = -S_{12} = \frac{-2r}{r^2 + 1}
\]

\[
r = \frac{\text{Ratio}}{Z_0}
\]
Hybrid90 (Ideal 90-degree Hybrid Coupler)

Symbol

Parameters

Loss = insertion loss, in dB
GainBal = gain balance between output ports, in dB
PhaseBal = phase balance between output ports, in degrees
Rref = port reference impedance, in specified units
Temp = temperature, in °C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
Delay = time delay for output ports

Range of Usage

Loss ≥ 0 dB
Delay ≥ 1.0e-16 sec

Notes/Equations

1. Although hybrid couplers are typically designed to give an equal power split between the coupled and the direct ports, the choice of coupling coefficients has been provided here for greater flexibility where a coupling of other than 3 dB is required.

2. The loss term is applied to both the coupling and the direct transmission coefficients.
Passive System Components

3. Pin designations (directional coupler notation/0-90 notation):
   (IN) = input port #1
   (ISO) = isolated output/input port #2
   (0) = coupled output / in-phase output
   (-90) = direct output/90-degree phase output

\[ |S_{0,IN}|(dB) = -10 \times \log \left(1 + 10^{-\text{GainBal}/10}\right) - \text{Loss} \]

\[ |S_{-90,IN}|(dB) = -10 \times \log \left(1 + 10^{-\text{GainBal}/10}\right) - \text{Loss} - \text{GainBal} \]

\[ |S_{0,ISO}|(dB) = -10 \times \log \left(1 + 10^{-\text{GainBal}/10}\right) - \text{Loss} - \text{GainBal} \]

\[ |S_{-90,ISO}|(dB) = -10 \times \log \left(1 + 10^{-\text{GainBal}/10}\right) - \text{Loss} \]

\[ \text{phase}(S_{0,IN}) = -360 \times \text{Delay} \times \text{frequency}(\text{degrees}) \]

\[ \text{phase}(S_{-90,IN}) = -360 \times \text{Delay} \times \text{frequency} - 90 - \text{PhaseBal}(\text{degrees}) \]

\[ \text{phase}(S_{0,ISO}) = -360 \times \text{Delay} \times \text{frequency} - 90 - \text{PhaseBal}(\text{degrees}) \]

\[ \text{phase}(S_{-90,ISO}) = -360 \times \text{Delay} \times \text{frequency}(\text{degrees}) \]

4. In general, Hybrid90 is not recommended for time-domain simulation. For the typical case of PhaseBal=0, time domain simulation will not produce proper results for \( S_{90,IN} \) or \( S_{0,ISO} \) because \( () \) in the frequency domain corresponds to a non-causal impulse response in time domain.

5. This component has no default artwork associated with it.
Hybrid180 (Ideal 180-degree Hybrid Coupler)

Symbol

Parameters
Loss = insertion loss, in dB
GainBal = gain balance between output ports, in dB
PhaseBal = phase balance between output ports, in degrees
Rref = port reference impedance, in specified units
Temp = temperature, in °C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
Delay = time delay for output ports

Range of Usage
Loss ≥ 0 dB
Delay ≥ 1.0e-16 sec

Notes/Equations
1. Although hybrid couplers are typically designed to give an equal power split between the coupled and the direct ports, the choice of coupling coefficient has been provided here for greater flexibility where a coupling of other than 3 dB is required.
2. The loss term is applied to both the coupling and the direct transmission coefficients.
3. Pin designations (directional coupler notation/sum-difference notation):
   (IN) = input port/input port #1
   (ISO) = isolated output/input port #2
   (Σ) = coupled output/summation output
Passive System Components

\[(\Delta) = \text{direct output/difference output}\]

\[|S_{\Sigma, \text{IN}}| \text{(dB)} = -10 \times \log(1 + 10^{(-\text{GainBal}/10)}) - \text{Loss} - \text{GainBal}\]

\[|S_{\Delta, \text{IN}}| \text{(dB)} = -10 \times \log(1 + 10^{(-\text{GainBal}/10)}) - \text{Loss}\]

\[|S_{\Sigma, \text{ISO}}| \text{(dB)} = -10 \times \log(1 + 10^{(-\text{GainBal}/10)}) - \text{Loss} - \text{GainBal}\]

\[|S_{\Delta, \text{ISO}}| \text{(dB)} = -10 \times \log(1 + 10^{(-\text{GainBal}/10)}) - \text{Loss} - \text{GainBal}\]

\[
\begin{align*}
\text{phase}(S_{\Sigma, \text{IN}}) &= -360 \times \text{Delay} \times \text{frequency} - \text{PhaseBal} \text{ (degrees)} \\
\text{phase}(S_{\Delta, \text{IN}}) &= -360 \times \text{Delay} \times \text{frequency} \text{ (degrees)} \\
\text{phase}(S_{\Sigma, \text{ISO}}) &= -360 \times \text{Delay} \times \text{frequency} \text{ (degrees)} \\
\text{phase}(S_{\Delta, \text{ISO}}) &= -360 \times \text{Delay} \times \text{frequency} - \text{PhaseBal} + 180 \text{ (degrees)}
\end{align*}
\]

4. When used in time domain simulation, set PhaseBal=0; for any PhaseBal value other than n x 180, time domain simulation will not produce proper results because the frequency domain specification corresponds to a non-causal impulse response in time domain.

5. This component has no default artwork associated with it.
IsolatorSML (SMLIsolator)

Symbol

Parameters
F1 = first frequency breakpoint
F2 = second frequency breakpoint
F3 = third frequency breakpoint
Loss1 = attenuation for frequencies ≤ F1, in dB
Loss2 = attenuation for frequencies > F1 ≤ F2, in dB
Loss3 = attenuation for frequencies > F2 ≤ F3, in dB
Loss4 = attenuation for frequencies > F3, in dB
VSWR1 = VSWR at both ports for frequencies ≤ F1
VSWR2 = VSWR at both ports for frequencies > F1, ≤ F2
VSWR3 = VSWR at both ports for frequencies > F2, ≤ F3
VSWR4 = VSWR at both ports for frequencies > F3
Isolat = isolation, in dB
Z1 = reference impedance for port 1
Z2 = reference impedance for port 2
Temp = temperature, in degrees C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations
1. All ports are assumed to have the same VSWR.
2. Isolation is assumed to be constant across all frequency bands.
Passive System Components

3. Up to three frequency breakpoints can be used to define four bands, with different losses and VSWR for each frequency band. If frequencies are not specified, this model is frequency independent and uses Loss1 and VSWR1 only.
LOS_Link (Line-Of-Sight Antenna Link)

Symbol

Parameters
CenterFreq = link center frequency, in hertz
BW = link bandwidth, in hertz
TxGain = transmitting antenna gain, in dB
TxVSWR = transmitting antenna VSWR
TxParabolaD = transmitting parabolic antenna diameter, in specified units
TxEfficiency = transmitting parabolic antenna efficiency
RxGain = receiving antenna gain, in dB
RxVSWR = receiving antenna VSWR
RxParabolaD = receiving parabolic antenna diameter, in specified units
RxEfficiency = receiving parabolic antenna efficiency
RxNoiseTemp = receiving antenna noise temperature, in Kelvin
PathLength = line-of-sight path length, in distance units
NotchFreq = notch frequency due to ground reflection path interference, in hertz
NotchDepth = power of reflected ray relative to direct ray, in dB
DeltaDelay = time delay of ground reflection path w.r.t. LOS path, in seconds
Z1 = transmitting antenna reference impedance, in ohms
Z2 = transmitting antenna reference impedance, in ohms

Range of Usage
PathLength > 10 wavelengths

Notes/Equations
1. Pathloss is infinite outside a window defined by BW.
Passive System Components

2. Transmitting antenna gain can be defined by TxGain or by TxParabolaD and TxEfficiency. If all three parameters are specified, TxParabolaD and TxEfficiency will overwrite TxGain. The same applies to receiving antenna as well.

3. The S-Parameter and noise implementation of this model is as follows. S11, S22, S12, S21, and I_Noise are derived from the user-specified LOS_link parameters.
Pad (Pi or Tee Format)

Symbol

Parameters

NetType = network type: Pi or Tee
Loss = attenuation, in dB
R1 = reference resistance for port 1
R2 = reference resistance for port 2
Temp = temperature, in degrees C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Range of Usage

R1 > 0
R2 > 0

Notes/Equations

1. This 2-port component provides:
   - Match to resistance $R_1$ at port 1 when port 2 is terminated in $R_2$
   - Match to $R_2$ at port 2 when port 1 is terminated in $R_1$
   - Desired attenuation, DB

2. Resistive networks can achieve simultaneous match at input and output if the insertion loss factor $A$ exceeds a minimum value given by

\[
A_{\text{min}} = 2 \times \frac{R_1}{R_2} \times \left(1 + \sqrt{\frac{R_2}{R_1}}\right) - 1.0 \quad R_1 > R_2
\]

\[
2 \times \frac{R_2}{R_1} \times \left(1 + \sqrt{\frac{R_1}{R_2}}\right) - 1.0 \quad R_2 > R_1
\]
where
\[
DB_{\text{MIN}} = 10 \times \log_{10} (A_{\text{MIN}})
\]

3. The values shown in the Pi network equivalent circuit are given by:

\[
\begin{align*}
G_c &= 2 \times \frac{\sqrt{A}}{(A - 1) \times \sqrt{R_1 \times R_2}} \\
G_a &= \frac{A + 1}{(A - 1) \times Z_1} - G_c \\
G_b &= \frac{A + 1}{(A - 1) \times R_2} - G_c
\end{align*}
\]

The values shown in the Tee network circuits are given by

\[
\begin{align*}
R_c &= \frac{2 \times \sqrt{A} \times \sqrt{R_1 \times R_2}}{(A - 1)} \\
R_a &= \frac{R_1 \times (A + 1)}{A - 1} - R_c \\
R_b &= \frac{R_2 \times (A + 1)}{A - 1} - R_c
\end{align*}
\]

Equivalent Circuit
PhaseShiftSML (Phase Shifter)

Symbol

Parameters
Phase = constant phase shift, in specified units
PhaseSlope = phase slope per frequency octave
FreqStart = frequency where slope begins
RTConj = reverse transmission conjugate: NO, YES
ZRef = reference impedance for all ports

Range of Usage
FreqStart ≥ 0

Notes/Equations
1. The output frequency spectrum is equal to the input spectrum shifted by the specified phase.
2. \( \theta(f) = \text{Phase} \) (for freq < FreqStart)
   \( \theta(f) = \text{Phase} + \text{PhaseSlope} \times \log_2 \left( \frac{\text{freq}}{\text{FreqStart}} \right) \) (for freq ≥ FreqStart)
   if FreqStart = 0, then phase slope is 0 regardless of the PhaseSlope setting
   where freq = simulation frequency
3. \( \frac{V_2}{V_1} = \frac{I_2}{I_1} = e^{j\theta(f)} \)
   \( S_{11} = S_{22} = 0 \)
   \( S_{21} = e^{j\theta(f)} \)
   if RTConj = NO, \( S_{12} = S_{21} \)
   if RTConj = YES, \( S_{12} = S_{21}^* \)
4. The S-parameter implementation for the phase shifter is shown here:
Passive System Components

5. In general, PhaseShiftSML is not recommended for time domain simulation. For the special case of $n \times 180^\circ$ phase shift, time domain simulation produces proper results; for the extreme case of $n \times 90^\circ$ phase shift, time domain simulation will not produce proper results because $(\pm)j$ in the frequency domain corresponds to a non-causal impulse response in time domain.
**PwrSplit2 (2-Way Power Splitter)**

**Symbol**

![Diagram of PwrSplit2](image)

**Parameters**

- $S_{21} =$ port 1 to port 2 complex transmission coefficient
- $S_{31} =$ port 1 to port 3 complex transmission coefficient
- $S_{11} =$ port 1 complex reflection coefficient
- $S_{22} =$ port 2 complex reflection coefficient
- Isolation = isolation between port 2 and port 3, in dB
- $Z_{Ref} =$ reference impedance for all ports
- Temp = temperature, in degrees C
- CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
- Delay = time delay, in seconds

**Notes/Equations**

1. $S_{12} = S_{21}, S_{13} = S_{31}, S_{33} = S_{22}$
2. PwrSplit2 can also be used as a power combiner.
3. Use the functions `polar()`, `dbpolar()` to represent the S-parameters in terms of magnitude, phase or dB, phase.
Passive System Components

**PwrSplit3 (3-Way Power Splitter)**

**Symbol**

![Diagram of PwrSplit3](image)

**Parameters**

- $S_{21} =$ port 1 to port 2 complex transmission coefficient
- $S_{31} =$ port 1 to port 3 complex transmission coefficient
- $S_{41} =$ port 1 to port 4 complex transmission coefficient
- $Z_{\text{Ref}} =$ reference impedance for all ports
- $\text{Temp} =$ temperature, in degrees C
- $\text{CheckPassivity} =$ check passivity flag; if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
- $\text{Delay} =$ time delay, in seconds

**Notes/Equations**

1. Ideal isolation exists for $S_{23}$, $S_{24}$, $S_{32}$, $S_{34}$, $S_{42}$, and $S_{43}$, that is, $S_{23} = S_{32} = S_{24} = S_{34} = S_{42} = S_{43} = 0$.
2. Ideal match exists for $S_{11}$, $S_{22}$, $S_{33}$, and $S_{44}$, that is, $S_{11} = S_{22} = S_{33} = S_{44} = 0$.
3. $S_{12} = S_{21}$, $S_{13} = S_{31}$ and $S_{14} = S_{41}$.
4. PwrSplit3 can also be used as a power combiner.
5. Use the functions `polar()`, `dBpolar()` to represent the S-parameters in terms of magnitude, phase or dB, phase.
**TimeDelay (Time Delay)**

**Symbol**

![Symbol Diagram](image)

**Parameters**

- **Delay** = time delay, in seconds (default = 1 µsec)
- **RTConj** = reverse transmission conjugate: NO (default), YES
- **ZRef** = reference impedance for all ports, in ohms (default = 50. Ohm)

**Range of Usage**

Delay ≥ 0

**Notes/Equations**

1. The input frequency spectrum has a linear phase shift applied resulting in the output time waveform being a time shifted replication of the input waveform.

2. $S_{21} = e^{-j\omega \text{Delay}}$

   $S_{11} = S_{22} = 0$

   $\omega = 2\pi f$

   if RTConj = NO, $S_{12} = S_{21}$

   if RTConj = YES, $S_{12} = S_{21}^*$

3. Excessive values for the Delay parameter require large memory for Circuit Envelope and Transient simulations. For Circuit Envelope simulations, a safeguard limits any excessive delay to one-half of the stop time.
Passive System Components

Transformer (Ideal 4-Port Transformer)

Symbol

Parameters
None

Range of Usage
N ≠ 0

Notes/Equations

Note  This component is obsolete for new designs. (It is available only for compatibility with designs created with 2002C or earlier releases.) Please use the TF transformer (Lumped-Components library) for new design work.

1. Input pins are 1 and 3; output pins are 2 and 4.
2. The S-parameters of the component are determined as follows:
   \[
   S_{11} = S_{24} = S_{33} = S_{42} = \frac{N^2}{1 + N^2}
   \]
   \[
   S_{14} = S_{23} = S_{32} = S_{41} = -\frac{N}{1 + N^2}
   \]
   \[
   S_{12} = S_{21} = S_{34} = S_{43} = \frac{N}{1 + N^2}
   \]
   \[
   S_{13} = S_{22} = S_{31} = S_{44} = \frac{1}{1 + N^2}
   \]
3. The input resistance R_{IN} output resistance R_{OUT} are related to N by:
   \[
   \sqrt{\frac{R_{IN}}{R_{OUT}}} = N
   \]
4. This is a noiseless component.
5. Because it is an ideal transformer, the impedance transformation is the same at DC as it is at nonzero frequencies.
6. This component passes DC.
TransformerG (Transformer with Ground Reference)

Symbol

Parameters
None

Range of Usage
N ≠ 0

Notes/Equations

Note This component is obsolete for new designs. (It is available only for compatibility with designs created with 2002C or earlier releases.) Please use the TF transformer (Lumped-Components library) for new design work.

1. The input resistance $R_{IN}$ and $R_{OUT}$ are related to N by:

$$\sqrt{\frac{R_{IN}}{R_{OUT}}} = N$$
Passive System Components

**TwoPort (2-Port Model)**

**Symbol**

![Symbol](image)

**Parameters**

- $S_{21} =$ complex forward transmission coefficient
- $S_{12} =$ complex reverse transmission coefficient
- $S_{11} =$ port 1 complex reflection coefficient
- $S_{22} =$ port 2 complex reflection coefficient
- $Z_{Ref} =$ reference impedance for all ports
- $Temp =$ temperature, in degrees C
- $CheckPassivity =$ check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.
- $Delay =$ time delay, in seconds

**Notes/Equations**

1. Use the functions `polar()`, `dBpolar()` to represent the $S$-parameters in terms of magnitude, phase or dB, phase.
Chapter 5: Phase Lock Loop Components
Phase Lock Loop Components

**DivideByN (Divide by N)**

**Symbol**

![Symbol Diagram]

**Parameters**

- **FnomIn** = nominal input frequency, in hertz
- **N** = divide number

**Notes/Equations**

1. This model performs a divide-by-N function on either a baseband-input or a selected-carrier input frequency. The model operates in transient, harmonic balance, or circuit envelope simulation.

   In transient, all signals are considered baseband.

   In circuit envelope, the **FnomIn** parameter defines which analysis frequency to use. If the analysis frequency is not within \(0.5/\text{timestep}\) a warning is issued and 0 Hz will be used for the analysis frequency.

2. The input impedance is infinite. The output impedance fixed at 1 ohm. If the output is in baseband mode, then its open circuit voltage is equal to the phase of the divided signal in radians. This results in a sawtooth waveform for a CW input. A sawtooth is output (instead of a square wave) to allow for the calculation of the actual frequency from the slope and for the accurate detection of zero crossings without the sampling jitter that would be introduced by the fixed rate sampling of a square wave. If the output is not in baseband mode, then the open circuit voltage is a 1V complex sinusoid at the divided frequency. Whether or not the output is in baseband is determined, in Circuit Envelope mode, by determining the carrier frequency, possibly dc, that is closest to **FnomIn**; if none are close enough, a warning is issued.

3. In non-baseband mode of operation, the divider works by directly extracting the phase of the complex input. The delay of DivideByN = \(\text{max}(\text{timestep, } N/(2 \times \text{Fin}))\), where Fin is the actual analysis frequency corresponding to **FnomIn**.

   In baseband mode of operation, the period of the input is determined by detecting when the baseband signal rises through the 0.5V threshold. From this
period information, the phase of the divided signal is calculated. For a more standard counter model, see the counter model under probes.

4. **Figure 5-1** show examples of the divider operating with an RF carrier input: one divider outputs a baseband output; one divider outputs the divided frequency at the other carrier frequency. Note that N can be time varying in order to simulate the effect of a fractional-N divider. **Figure 5-2** shows the simulation results.
Phase Lock Loop Components

Figure 5-2. Output Waveforms
**PhaseFreqDet (Frequency Detector, Baseband)**

**Symbol**

```
+----------------+  +----------------+
|                |  |                |
| PHASE/FREQ     |  |                 |
|  ^            |  |                 |
|  |            |  |                 |
|  |            |  |                 |
|  +-----------+  | +-----------+    |
|  |            |  |            |    |
|  |            |  |            |    |
|  |            |  | 01         | 02|

```

**Parameters**

Vhigh = High-state output voltage

Vlow = Low-state output voltage

**Notes/Equations**

1. PhaseFreqDet is not selectable from the component palette or component library browser; to place this component, type its exact name into the Component History box above the drawing area, then move your cursor to the drawing area.

2. This baseband phase-frequency demodulator is used in transient or circuit envelope simulation. It models the digital behavior of common D flip-flop type phase-frequency detectors often used in phase-locked loops. The two outputs are ideal, zero impedance voltage sources; to model a pulsed current-source output, two VDCS-dependent sources must be added to the output of this model. The two inputs have infinite impedance, and only the baseband portion of the two input voltages are used to determine threshold-crossing timing.

3. As opposed to the tuned phase-frequency detector model, this model’s output includes the effect of reference clock feed-through.

---

**Note**

The output of this model is a pulse train whose average value is proportional to the input phase difference, and may contain significant signal energy at the reference clock rate and at clock harmonics. These must be filtered out, typically before driving a VCO in a PLL application. The tuned phase-frequency detector output signal includes the instantaneous phase difference information only; it does not contain reference frequency or harmonic content.

However, the penalty for this is that the timestep must be less than one-half the reference period, and typically less than one-tenth the period. To avoid the large
Phase Lock Loop Components

amount of time jitter and phase noise that would normally be introduced by sampling at even these rates, the two digital outputs are also amplitude modulated to reflect the portion of a simulation timestep where the actual outputs would be high or low. For example, if based on the threshold-crossing timing (the pulse width should be 10 nsec, but the simulation timestep is 1\µsec), then the output amplitude for that timestep would only be 1% of the V\text{high} level. While this will not properly model all the higher harmonics of the reference feed-through, it does accurately model both the dc term and the first few harmonics, and the corresponding reference sidebands of the VCO. As the analysis timestep is further reduced, the behavior becomes more truly digital in nature and more harmonics are effectively being simulated at the cost of slower simulations.

4. The trigger times for both inputs is determined by detecting when the baseband voltage rises through the 0.5V threshold. Linear interpolation is used to get much finer time resolution than the analysis timestep. To further reduce excessive sampling jitter, the inputs to this detector can be sawtooth waveforms. While the detector will work reasonably well with sinusoidal inputs, given a small enough timestep, timing jitter can be eliminated if the interpolation is done on the positive slope of a sawtooth waveform. This is the reason why the divide-by-N models output a sawtooth waveform when they operate in the baseband mode. Square-wave inputs should generally be avoided, because this will usually introduce significant timing jitter and phase noise into the simulation.

5. This model does not include any effects due to the finite duration of the flip-flop reset pulse and resultant zero-phase dead zone. Other effects, such as asymmetry between the two different outputs, can be incorporated by changing the external components. Figure 5-3 shows an example using this detector; Figure 5-4 shows the output waveforms.
Figure 5-3. PhaseFreqDet Example
Figure 5-4. Output Waveforms
PhaseFreqDet2 (Frequency Detector, Baseband)

Symbol

Parameters

- \( V_{\text{high}} \) = high-state output voltage
- \( V_{\text{low}} \) = low-state output voltage
- \( \text{DeadTime} \) = dead zone pulse width
- \( \text{Jitter} \) = input time jitter

Notes/Equations

1. This baseband phase-frequency demodulator is used in transient or circuit envelope simulation. It models the digital behavior of common D flip-flop type phase-frequency detectors often used in phase-locked loops. The four outputs are ideal, zero impedance voltage sources. The two inputs have infinite impedance, and only the baseband portion of the two input voltages are used to determine threshold-crossing timing.

2. The following FDD modeling equations illustrate how the shape of the four outputs relate to the input phase difference.

The phase difference between the input on port 1 and port 2 is calculated as

\[ ns = \text{phase_freq}(1, 2) \]
and the quantities \( n1 = (V_{\text{high}} - V_{\text{low}}) \times \text{real}(ns) \) and \( n2 = (V_{\text{high}} - V_{\text{low}}) \times \text{imag}(ns) \) are introduced.

Outputs on ports 3-6 are then

- port 3: \( Q1 = V_{\text{low}} + n1 \)
- port 4: \( Q2_{\text{bar}} = V_{\text{high}} - n2 \)
- port 5: \( Q1_{\text{bar}} = V_{\text{high}} - n1 \) (only PhaseFreqDet2, not PhaseFreqDet)
- port 6: \( Q2 = V_{\text{low}} + n2 \) (only PhaseFreqDet2, not PhaseFreqDet)

with the following trigger events:
- \( \text{Trig}[1] = _{\text{xcross}} (1, 0.5, 1) \)
- \( \text{Trig}[2] = _{\text{xcross}} (3, 0.5, 1) \)
Phase Lock Loop Components

For details about these functions, refer to the FDD device documentation in the Circuit Components Nonlinear Devices manual.

Note that you can push into the component to see the implementation of the component.

3. As opposed to the tuned phase-frequency detector model, this model's output includes reference clock feed-through effects.

**Note** The output of this model is a pulse train whose average value is proportional to the input phase difference, and may contain significant signal energy at the reference clock rate, and at clock harmonics. Typically, these must be filtered out before driving a VCO in a PLL application. The tuned phase-frequency detector output signal includes instantaneous phase difference information only (it does not contain reference frequency or harmonic content).

However, the penalty for this is that the timestep must be less than one-half the reference period, and typically less than one-tenth the period. To avoid the large amount of time jitter and phase noise that would normally be introduced by sampling at even these rates, the four digital outputs are also amplitude modulated to reflect the portion of a simulation timestep where the actual outputs would be high or low. For example, if based on the threshold-crossing timing (the pulse width should be 10 nsec, but the simulation timestep is 1µsec), then the output amplitude for that timestep would only be 1% of the Vhigh level. While this will not properly model all the higher harmonics of the reference feed-through, it does accurately model both the dc term and the first few harmonics, and the corresponding reference sidebands of the VCO. As the analysis timestep is further reduced, the behavior becomes more truly digital in nature and more harmonics are effectively being simulated at the cost of slower simulations.

4. The trigger times for both inputs is determined by detecting when the baseband voltage rises through the 0.5V threshold. Linear interpolation is used to get much finer time resolution than the analysis timestep. To further reduce excessive sampling jitter, inputs to this detector can be sawtooth waveforms. While the detector will work reasonably well with sinusoidal inputs, given a small enough timestep, the timing jitter can be eliminated if the interpolation is done on the positive slope of a sawtooth waveform. This is the reason why the divide-by-N models output a sawtooth waveform when they operate in the baseband mode. Square wave-inputs should generally be avoided, because this

5-10 PhaseFreqDet2 (Frequency Detector, Baseband)
will usually introduce significant timing jitter and phase noise into the simulation.

5. **DeadTime** specifies the period of time centered around the 0 phase output during which no output is generated from either the high or the low charge pump. Outside of this period, the output returns to the ideal pulse widths and amplitudes determined by the trigger crossings of the inputs and the charge pump currents.

6. The **Jitter value** defines the RMS time jitter associated with a trigger crossing on either input. The distribution is Gaussian and the noise spectrum is assumed flat out to the reference sampling frequency.

7. Other effects, such as asymmetry between the four different outputs, can be incorporated by changing the external components. **Figure 5-5** shows one application of this detector and **Figure 5-6** the resultant output waveforms.

![Figure 5-5. One application of detector](image-url)
Figure 5-6. Resultant output waveforms of detector
PhaseFreqDetCP (Frequency Detector, Baseband with Charge Pump)

Symbol

Parameters

I\text{high} = \text{high-level charge pump current}

I\text{low} = \text{low-level charge pump current}

DeadTime = \text{dead-zone pulse width}

Jitter = \text{input time jitter}

Notes/Equations

1. This baseband phase-frequency demodulator with charge pump is used in transient or circuit envelope simulation.

2. Charge pump currents can be constants (in which case they are ideal current sources) or functions of the state variable \( v_2 \) to allow for a non-ideal current whose peak value varies as a function of the output voltages. In either case, the off-state current is always 0.0. Normally positive values would be used for both currents, as \( I_{\text{high}} \) specifies a source current and \( I_{\text{low}} \) specifies the sink current.

3. DeadTime specifies the period of time centered around the 0 phase output during which no output is generated from either the high or the low charge pump. Outside of this period, the output returns to the ideal pulse widths and amplitudes determined by the trigger crossings of the inputs and the charge pump currents.

4. The Jitter value defines the RMS time jitter associated with the trigger crossing at each of the two inputs. The effective output time jitter will be \( \sqrt{2} \) times this value. The distribution is Gaussian and the noise spectrum is assumed flat out to the reference sampling frequency.
Phase Lock Loop Components

**PhaseFreqDetTuned (Phase Frequency Detector, Tuned)**

**Symbol**

```
1
2
3
```

**Parameters**

- Sensitivity = detector sensitivity, in mA/degree
- MaxAngle = maximum unwrapped phase angle, in degrees
- Vlimit = maximum output voltage compliance, in volts
- Fnom = nominal input frequency for VCO and REF inputs, in hertz

**Notes/Equations**

1. This tuned phase-frequency demodulator is used with circuit envelope simulation that models the ideal behavior of the phase frequency detectors used in phase-locked loops. (This model does not work with transient simulation.)

   On the REF and the VCO inputs, it selects the input carrier closest to the specified Fnom frequency. For proper operation, this cannot be the baseband (dc) envelope. It then generates a baseband output signal equal to the phase difference between the VCO and REF inputs, with an offset of $2 \times \pi \times \Delta f \times t$ being generated if there is a frequency difference $\Delta f$ between the two inputs.

   As an example consider the sources $V_1(t) = \exp(j \times P_1(t))$ at $f_0$ and $V_2(t) = \exp(j \times P_2(t))$ also at $f_0$, as input to a PhaseFreqDetTuned component with the output terminated in a high impedance. Envelope simulation at $f_0$ yields a DC output with a $P_1(t) - P_2(t)$ time dependence. Changing the frequency of the second source from $f_0$ to $f_0 + \Delta f$ for the same $V_2(t)$ is equivalent with keeping the frequency at $f_0$ but multiplying $V_2(t)$ by $\exp(j \times 2 \times \pi \times \Delta f \times t)$ and results in the output signal $P_1(t) - P_2(t) - (2 \times \pi \times \Delta f \times t)$.

   If there is no analysis harmonic frequency close enough to the Fnom frequency, then a warning is generated. The two input impedances are infinite. The output is a baseband-only infinite impedance current source with scaling defined by the Sensitivity parameter. If the output voltage exceeds Vlimit or is lower than $-Vlimit$, then a voltage limiter with an impedance of 10 ohms is switched in.

   The MaxAngle parameter can be used to model special phase-frequency detectors that have greater than the traditional $\pm 2\pi$ of input phase range.
2. This model does not model any of the reference or VCO frequency components on the output. The output is a baseband-only signal, and the voltage limiting applies to the baseband component only. (See the baseband phase-frequency detector for a model that can overcome this limitation.) With this tuned model, though, the timesteps can actually be larger than the reference period, and it is only dependent on the PLL bandwidth and not the reference frequency; this enables faster performance for many types of simulation measurements.

Another limitation is that only the phase of the selected input envelope is included. The effect of additional input spurs and harmonics are not included if they are in different carrier frequency envelopes.

3. Figure 5-7 and Figure 5-8 show a simple schematic and example of this phase-frequency detector's output. The reference input is a fixed frequency. The VCO input has sinusoidal frequency modulation.

![Figure 5-7. Phase-Frequency Detector Example](image-url)
Phase Lock Loop Components

Figure 5-8.
PhaseNoiseMod (Phase Noise Modulator)

Symbol

Parameters

- $F_{\text{nom}}$ = nominal input frequency, in hertz
- $R_{\text{out}}$ = output resistance, in ohms
- $F_{\text{corner}}$ = corner frequency for 1/f noise performance, in hertz
- $\text{NF}$ = broadband noise figure, in dB
- $Q_{L}$ = loaded Q of resonator

Notes/Equations

1. This device uses Leeson’s equation to model oscillator phase noise, then modulates the input carrier with this phase noise.

The input can be from any signal source, including the VCO models. This model behaves as a tuned modulator by selecting and modulating just the carrier defined by the $F_{\text{nom}}$ parameter. If there are no analysis frequencies close enough to this value, a warning is generated and no signal is output. The Leeson’s equation models the oscillator phase using the equation

$$S_{\Delta \phi} = \frac{(F)kTB}{P_{\text{sigAV}}} \times \left(1 + \frac{F_{\text{nom}}}{2Q_{L}F_{\text{req}}}\right)^{2} \times \left(1 + \frac{F_{\text{corner}}}{F_{\text{req}}}\right)$$

where

- $F_{\text{nom}}$ is used as an approximation to carrier frequency
- $F$ is the noise factor
- $P_{\text{sigAV}}$ is the input signal power
- $T$ is the absolute temperature
- $B$ is the analysis bandwidth
- $k$ is the Boltzmann constant
- $Q_{L}$ is the loaded Q value of the oscillator’s resonator
- $F_{\text{corner}}$ is the frequency at which the low frequency 1/f noise is equal to the broadband noise.
Phase Lock Loop Components

This model is usable in frequency domain and circuit envelope time domain noise analyses. To avoid the divide-by-zero problems as the analysis offset frequency approaches 0, both the $1/f^2$ and $1/f$ terms are rolled off at frequencies below 1 Hz. In the time-domain mode, the $1/f$ frequency response is implemented by doing a convolution simulation. The duration of this impulse response is set to 2000 timesteps. This effectively rolls off this $1/f$ response at a frequency determined by the analysis tstep parameter.
VCO (Voltage Controlled Oscillator)

Symbol

Parameters

$K_v =$ frequency tuning sensitivity, in hertz per volt
$Freq =$ fundamental frequency, in hertz
$P =$ power into $Rout$ load at fundamental frequency
$Rout =$ output resistance, in ohms
$Delay =$ transit time delay added to input tuning voltage, in seconds
$Harmonics =$ ratio of harmonic voltage to fundamental voltage, complex value

Notes/Equations

1. This is a simple VCO model that outputs a signal whose frequency is controlled in a linear manner by the input tuning voltage. The center frequency is defined by the $Freq$ parameter, which references one of the analysis frequencies. A $Freq$ value of 0 references dc and so defines the VCO as a baseband source; the output frequency is the baseband input voltage times the $K_v$ parameter value. Only the baseband portion of the input tuning voltage is used to determine the VCO frequency offset.

2. The phase of the VCO output is clamped when time equals 0, so this model only functions as a VCO in the time-domain analysis modes, including circuit envelope and transient simulation. Because transient simulation is a baseband-only analysis mode, $Freq$ should be 0 in this mode.

3. The output resistance at all frequencies is set by $Rout$, which is internally limited to a minimum value of 0.1 ohm. The VCO's fundamental output power into an $Rout$ load is defined by the $P$ parameter. The relative level, in linear units, of the second and third harmonics are defined by the $Harmonics$ parameter; this parameter can be complex to allow definition of the relative phase of these harmonics. The initial phase for the fundamental frequency output is set to $-90$ degrees so that a sine waveform is created in baseband mode.
4. A Delay parameter value can also be specified for this VCO model. This puts an additional transit delay between the input tuning voltage and the actual change in the output frequency. A delay of at least one time-step does sometimes result in faster simulation speeds, and can be used to model the time delay inherent in any real VCO.

Care should be taken when using the VCO in baseband mode (Freq = 0). If the input voltage is allowed to go negative, the model will generate a negative frequency, and the waveform shape due to the harmonic content will be reversed.

The VCO_DiviDeBYn model also allows the definition of a nonlinear frequency tuning characteristic.

5. Figure 5-9 shows the use of the VCO in a transient simulation. The VCO is defined with significant second and third harmonic levels. Simulation results are shown in Figure 5-10.
Figure 5-10. Simulation Results
Phase Lock Loop Components

VCO_DivideByN (VCO Divide By N)

Symbol

Parameters
VCO_Freq = frequency deviation from F0 (function of _v1), in hertz per volt
F0 = VCO center frequency, in hertz
N = nominal divide number (with dN=0)
Rout = output resistance of VCO, in ohms
Power = output power into Rout load, in watts
Delay = transit time delay added to input tuning voltage, in seconds

Pin Connections
- tune connects to a tuning voltage.
- dN connects to ground (dN=0) or a voltage source such as V_DC.
- dN takes the value of nodal voltage and N+dN becomes the divide ratio.
- vcon is the divided-by-(N+dN) output.
- freq outputs the undivided frequency values from pin VCO. This pin can be left open.
- VCO is the undivided VCO output. This pin can be left open if an undivided VCO is not used.

Notes/Equations
1. This VCO model allows for the definition of an arbitrary, nonlinear frequency tuning characteristic. In addition, it incorporates a behavioral, divide by N model. Incorporating the divider into the same model permits its use in phase-lock loop simulations where the envelope bandwidth, as determined by the analysis time-step, does not have to include the entire tuning range of the VCO, but only the frequency range of the divided output. In these cases, the phase and frequency information of the VCO’s main output may be aliased because of the large time-step. But, if just the divided output is being used, the
loop simulations will still be valid and can simulate faster due to the large time-step.

As the time-step is decreased, or the range of the VCO is reduced, such that it remains within the envelope bandwidth, then both the main VCO and the divided VCO output are valid. As with the standard VCO model, the phase of this model's outputs are also clamped when time=0 so this model only functions as a VCO in the time-domain analysis modes, including circuit envelope and transient simulations.

2. The frequency of the VCO is determined by the F0 value plus the VCO_Freq value. The VCO_Freq value may be an arbitrary expression using _v1, which is a pre-defined variable representing the input tuning voltage. The frequency is determined inside the model by determining the present value of _v1 (the input voltage), evaluating the VCO_Freq expression and adding this to the F0 value. This frequency is output, as an ideal voltage source scaled to 1V per GHz, on the freq output pin.

3. In circuit envelope simulation, the carrier frequency envelope associated with the main VCO frequency is determined by the F0 value. If there is no analysis harmonic frequency close enough to F0, then a warning is generated and the main output is zero. The carrier frequency envelope associated with the divided VCO frequency is determined by F0/N. Again, if no analysis harmonic frequency is close enough, then a warning is generated and this output is also set to zero.

4. The divide number is determined by adding the N parameter and the dN baseband input voltage. The divide number can change during the simulation. By properly driving the dN input, fractional frequency division can be simulated. To simulate all the dynamics of a fractional divider, the simulation time-step must be small enough to properly digitize the varying divide- or pulse-swallowing rate. Alternatively, either the N value or the dN input can be set to fractional values to obtain a steady-state, fractional division that would not include the switching dynamics and spurs. In circuit envelope simulation, the divided VCO frequency must remain within its initial envelope bandwidth for all combinations of VCO frequencies and divide numbers. It will not automatically jump from one envelope carrier frequency to another. In transient simulation, because everything is treated as baseband signals, the only constraint is that the time-step must be small enough to cover the maximum frequency.
5. Both the main VCO output and the divided output have an output resistance set by the Rout parameter. The main VCO output will deliver the specified Power into a Rout load. The divided output will also deliver this amount of power if it is not a baseband output. If it is a baseband output, then the divided output is a sawtooth waveform, whose open circuit voltage represents the instantaneous phase, in radians, of the divided signal. In transient simulation, then, this divided output is always a sawtooth. In circuit envelope, it is a sawtooth if F0/N is within the baseband envelope (it is less than 0.5/timestep). If F0/N is closer to one of the analysis carrier frequencies, then the output is a complex sinusoid with the same amplitude as the main VCO output.

6. A Delay parameter value can also be specified for this VCO model. This puts an additional transit delay between the input tuning voltage and the actual change in the output frequency. A delay of at least one timestep does sometimes result in slightly faster simulation speeds, and can be used to model the time delay inherent in any real VCO.

7. Care should be taken when using the VCO in baseband mode. If F0+VCO_Freq value goes negative, the model will generate a negative frequency, which may give unexpected results. If this is a problem, the VCO_Freq expression could include a limiting operator to prevent this.

8. Figure 5-11 shows an example application; simulation results are shown in Figure 5-12. The tuning characteristic is linear in this case and is simply 1 MHz per volt. The nominally divided output frequency is 100 MHz/55=1.81818 MHz, so this output can be a baseband output, given the 0.1 μsec timestep. The divide number changes from 95 to 55 halfway through the simulation. (Note that while the divided VCO output may not appear to be a regular, uniform, amplitude sawtooth waveform, it does accurately represent the divided signal's phase.)

9. The frequency can be determined by calculating the phase slope; by using linear interpolation, the baseband phase frequency detectors can accurately determine threshold crossings.
Figure 5-11. Example with Linear Tuning Characteristics

Figure 5-12. Simulation Results
Phase Lock Loop Components
Chapter 6: Switch and Algorithmic Components
Switch and Algorithmic Components

Comparator (Comparator)

Symbol

![Comparator Symbol]

Parameters

\( V_{\text{low}} = \) lower threshold voltage

\( V_{\text{high}} = \) upper threshold voltage

Notes/Equations

1. Comparator outputs a 1V signal whenever the baseband portion of the input signal is between the two threshold voltages. Input impedance is infinite; output impedance is fixed at 0.1 ohm; and, there is a fixed delay of one timestep.

2. If the baseband portion of \( V_{\text{in}} \) is greater than \( V_{\text{low}} \) and less than \( V_{\text{high}} \), then the output voltage at the next time sample is 1.0V; otherwise, the output is 0.0V.

3. This model works in transient and circuit envelope simulation.
ClockLFSR (Linear Feedback Shift Register)

Symbol

Parameters
Vlow = lower threshold voltage, in volts
Vhigh = upper threshold voltage, in volts
Taps = bits used to generate feedback
Seed = initial value loaded into shift register
Rout = output resistance, in ohms

Notes/Equations
1. ClockLFSR can be used to generate PN sequences with user-defined recurrence relations. The input is a clock signal; with each positive clock edge, the next output bit is calculated. A clock edge occurs any time the baseband input signal rises through 0.5V.

2. This model works in transient and circuit envelope simulation.

3. With each positive clock edge, data is shifted to the right in the shift register. The length of the shift register is determined by the most significant one-bit in the Taps value.

Figure 6-1. LFSR Model
Switch and Algorithmic Components

The numbers \(a(1), a(2), \ldots, a(r)\) are the binary feedback coefficients and are specified by the Taps parameter. This value may be specified as a decimal number, or as a binary pattern if the bin() function is used. The initial contents of the shift are specified by the value of the Seed parameter. The following equations describe the operation of this component:

At each positive clock edge \(n (n \geq 1)\), for \(n \geq 1\):

\[
D(n) = \left[ \sum_{k=1}^{r} a(k)D(n-k) \right] \mod 2
\]

where

\[
D(0) = \text{Seed}_2(0) \\
D(-1) = \text{Seed}_2(1) \\
\ldots \\
D(1-r) = \text{Seed}_2(r-1)
\]

and \(\text{Seed} = \sum_{k \geq 0} \text{Seed}_2(k)2^k\)

where \(\text{Seed}_2(k) \in \{0,1\}\) for \(0 \leq k < r\).

Example: Let \(\text{Seed} = 2\), and \(\text{Taps} = 7\) then

\[
\text{Seed}_2(0) = 0 \\
\text{Seed}_2(1) = 1 \\
\ldots \\
\text{Seed}_2(6) = 0
\]

thus,

\[
D(0) = \text{Seed}_2(0) = 0 \\
D(-1) = \text{Seed}_2(1) = 1 \\
D(-2) = \text{Seed}_2(2) = 0
\]
D(-6) = Seed2(2) = 0

Linear Feedback Shift Register ClockLFSR

The binary feedback coefficients are specified by Taps. For example, the recurrence relation

\[ D(n) = (D(n-7) + D(n-3) + D(n-2) + D(n-1)) \mod 2 \]

is specified by:

Taps = bin("1000111")

or

Taps = 71

Table 6-1 provides a list of feedback coefficients for linear feedback shift registers showing one or more alternate feedback connections for a given number of stages.

<table>
<thead>
<tr>
<th>No. of Stages</th>
<th>Code Length</th>
<th>Maximal Taps</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>[2, 1]</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>[3, 1]</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>[4, 1]</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>[5, 2, 1]</td>
</tr>
<tr>
<td>6</td>
<td>63</td>
<td>[6, 1]</td>
</tr>
<tr>
<td>7</td>
<td>127</td>
<td>[7, 1]</td>
</tr>
<tr>
<td>8</td>
<td>255</td>
<td>[8, 4, 3, 2]</td>
</tr>
<tr>
<td>9</td>
<td>511</td>
<td>[9, 4]</td>
</tr>
<tr>
<td>10</td>
<td>1023</td>
<td>[10, 3]</td>
</tr>
<tr>
<td>11</td>
<td>2047</td>
<td>[11, 5, 3, 1]</td>
</tr>
<tr>
<td>12</td>
<td>4095</td>
<td>[12, 6, 4, 1]</td>
</tr>
<tr>
<td>13</td>
<td>8191</td>
<td>[13, 4, 3, 1]</td>
</tr>
</tbody>
</table>

## Switch and Algorithmic Components

<table>
<thead>
<tr>
<th>No. of Stages</th>
<th>Code Length</th>
<th>Maximal Taps</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>16,383</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>15</td>
<td>32,767</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>16</td>
<td>65,535</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>17</td>
<td>131,071</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>18</td>
<td>262,143</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>19</td>
<td>524,287</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>20</td>
<td>1,048,575</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>21</td>
<td>2,097,151</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>22</td>
<td>4,194,303</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>23</td>
<td>8,388,607</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>24</td>
<td>16,777,215</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>25</td>
<td>33,554,431</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>26</td>
<td>67,108,863</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>27</td>
<td>134,217,272</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>28</td>
<td>268,435,455</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>29</td>
<td>536,870,911</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>30</td>
<td>1,073,741,823</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>31</td>
<td>2,147,483,647</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>32</td>
<td>4,294,967,295</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>33</td>
<td>8,589,834,591</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
<tr>
<td>34</td>
<td>17,178,669,279</td>
<td><img src="#" alt="Taps List" /></td>
</tr>
</tbody>
</table>


6-6 ClockLFSR (Linear Feedback Shift Register)
Differentiator (Differentiator)

Symbol

Parameters

Gain = differentiator slope

Rref = reference resistance for both ports, in ohms

Notes/Equations

1. The output voltage is equal to the derivative (with respect to time) of the input voltage.

2. The Gain parameter is entered as a linear quantity, not in dB.

3. Differentiator works in both transient and envelope simulations.

4. The input resistance of the integrator is Rref. The output is a voltage source Vout in series with output resistance Rref.
Switch and Algorithmic Components

**DPDT_Static (Double Pole Double Throw Switch, Static)**

**Parameters**

State = state of switch: 0 (nodes 1 and 2, 4 and 5 connected); 1 (nodes 1 and 3, 4 and 6 connected)

F1 = first frequency breakpoint
F2 = second frequency breakpoint
F3 = third frequency breakpoint
Loss1 = attenuation for frequencies ≤ F1, in dB
Loss2 = attenuation for frequencies > F1 ≤ F2, in dB
Loss3 = attenuation for frequencies > F2 ≤ F3, in dB
Loss4 = attenuation for frequencies > F3, in dB
VSWR1 = VSWR at both ports for frequencies ≤ F1
VSWR2 = VSWR at both ports for frequencies > F1 ≤ F2
VSWR3 = VSWR at both ports for frequencies > F2 ≤ F3
VSWR4 = VSWR at both ports for frequencies > F3
Isolat = isolation, in dB
ZRef = reference impedance for all ports, in ohms
Temp = temperature, in degree C
Notes/Equations

1. This model works in transient and circuit envelope simulation.

2. Up to three frequency break points can be used to define four bands, with different losses and VSWR for each frequency band. Losses are entered as a positive attenuation. In other words, a 0.5 dB loss is entered as $L_1=0.5$, not $L_1=-0.5$.

   Isolation is assumed to be constant across all frequency bands.

   When using this device in a transient simulation, the switch setting must be constant.

   If loss values are specified versus frequency via a dataset and using the component DATASET_VARIABLE, then a time delay through the device will be introduced when using transient analysis.

   Transient Analysis Time Delay = $\left(\frac{1}{2 \times \text{MAX\_FREQUENCY}}\right) \times \text{POINTS}$

where

   $\text{POINTS} = \left(\frac{\text{FILTER\_ORDER}}{2}\right) - 1$ for \text{FILTER\_ORDER} even

   $= \left(\frac{\text{FILTER\_ORDER} + 1}{2}\right) - 1$ for \text{FILTER\_ORDER} odd
Switch and Algorithmic Components

IntegratorSML (Integrator)

Symbol

Parameters

GainAC = integrator gain
GainDC = gain of dc constant
Rref = reference resistance for both ports, in ohms

Notes/Equations

1. The output voltage is equal to the integral (with respect to time) of the input voltage. This device has been modeled after a physically realizable integrator. The dc gain is not infinite and is specified by GainDC.

2. GainAC is the ac gain of the integrator (typically set to 1.0); GainDC is the dc gain of the integrator. Both of these are a linear quantity, not in dB.

3. An ideal integrator has infinite gain at dc, which is not physically realizable. These parameters describe the operation of the integrator that has been modeled after a traditional high gain differential amplifier with capacitive feedback and a series input resistor. GainDC is the open loop dc gain of the amplifier in this model. Gain is equal to 1/RC where R is the input series resistor value in ohms and C is the feedback capacitor value in farads.

4. The integrator transfer function follows:

\[
\frac{V_{out}}{V_{in}} = \frac{GainDC}{1 + (j \times w \times GainAC \times GainDC)}
\]

where \(w\) is the frequency in radians/sec.

5. IntegratorSML works in both transient and envelope simulations.

6. The input resistance of the integrator is Rref. The output is a voltage source Vout in series with output resistance Rref.
LimiterSML (Limiter)

Symbol

Parameters
Gain = linear gain
Vmax = maximum output voltage, in volts
Vmin = minimum output voltage, in volts
Rout = output resistance, in ohms
Fnom = nominal input frequency, in hertz

Range of Usage
Fnom ≥ 0

Notes/Equations
1. The Gain parameter is a real value only (not in dB or complex).
2. This model can be used in transient, harmonic balance and circuit envelope simulations.
3. When used in transient simulations, Fnom must be set to zero. Gain, Vmin and Vmax determines the output voltage time series. The output voltage time series is calculated as Gain times the input voltage time series but is hard limited to Vmin and Vmax in case the value drops below Vmin or rises above Vmax. Hard limiting is not allowed to drop the magnitude of the output voltage by more than a factor of 10^-4 at any point in time. If this happens, the input signal is simply scaled by Gain and 10^-4. Thus, a sine wave with 10^6V amplitude as input to a LimiterSML component with Gain=10, Vmax=1V and Vmin=-1V will scale to an output signal with a 10^3V amplitude, not a 1V amplitude. This signal will be added to a pulse function that is Vmin for negative input signals and Vmax for positive input signals so the output will not be a perfect sine wave.
4. When used in harmonic balance simulations, Fnom must be set to a non-zero fundamental or intermodulation frequency. Fnom, Gain and abs(Vmax) determines the output voltage spectrum (Vmin is not used). The output voltage spectrum is calculated as Gain times the input voltage spectrum but is scaled...
by \( \min(\frac{\text{abs}(V_{\text{max}})}{\text{abs}(V(F_{\text{nom}}))}) \) at all frequencies, where \( V(F_{\text{nom}}) = \text{Gain} \times V_{\text{in}}(F_{\text{nom}}) \). This gain limits the entire frequency spectrum by \( \frac{\text{abs}(V_{\text{max}})}{\text{abs}(V(F_{\text{nom}}))} \) in the case where \( \text{abs}(V(F_{\text{nom}})) \) exceeds \( \text{abs}(V_{\text{max}}) \). If there is no spectral component at \( F_{\text{nom}} \), no gain limiting is performed. No minimum gain of \( 10^{-4} \) applies in this case.

5. When used in circuit envelope simulations, \( F_{\text{nom}} \) can be a zero or non-zero fundamental or intermodulation frequency.

- If \( F_{\text{nom}} \) is zero, the mode of operation is similar to that for transient analysis. If there is a baseband signal, it is hard limited as dictated by \( V_{\text{min}} \) and \( V_{\text{max}} \). Hard limiting is not allowed to drop the magnitude of the output voltage by more than a factor of \( 10^{-4} \) at any time (refer to note 3). All other spectral components are scaled accordingly. If there is not a baseband signal, nothing happens.

- If \( F_{\text{nom}} \) is a non-zero fundamental or intermodulation frequency, the mode of operation is similar to that for harmonic balance analysis. All spectral components, including the baseband signal if applicable, are gain limited as dictated by \( \min(\frac{\text{abs}(V_{\text{max}})}{\text{abs}(V(F_{\text{nom}}))}) \). For harmonic balance analysis, the value of \( V_{\text{min}} \) does not matter in this case.

6. LimiterSML has more than one mode of operation. It performs clipping or limiting in a Transient or Circuit Envelope analysis, and it performs gain scaling on the spectral components in a Harmonic Balance analysis. Its mode of operation depends on the value of \( F_{\text{nom}} \) and the type of simulation being performed.
ParallelSerial (Parallel to Serial Shift Register)

Symbol

Parameters
OutputRate = serial output data clock rate, in hertz
LSB_First = output serial data with least significant bit first = YES; NO = output serial data with most significant bit first
Delay = initial synchronization delay
InputBits = number of bits in input word
IntegerIn = scale input data as integers: YES, NO

Notes/Equations
1. This Parallel To Serial Shift Register model is used to convert a sequence of input words into a serial output bit stream. The fixed serial output bit rate is specified by the OutputRate parameter. The serial data can be output with either the LSB or MSB first, depending on the state of the LSB_First parameter. The output impedance is fixed at 0.1 ohm. A logic one generates an open circuit voltage of 1.0V. A logic zero is 0.0V.
2. The number of bits in each input word is specified by InputBits. The input is sampled at a rate equal to OutputRate/InputBits, with an initial synchronization delay specified by Delay.
3. The input impedance is infinite. If the IntegerIn parameter is set true, then the input is assumed to be scaled as an integer from 0.0V to 2^{OutputRate-1}V. Otherwise, the input is assumed to be scaled from −1.0 to +1.0V, with −1.0V interpreted at word 0.
4. This model works in transient and circuit envelope simulation. Only the baseband portion of circuit envelope voltage is used.
Switch and Algorithmic Components

QuantizerSML (Quantizer)

Symbol

Parameters

- Vmin = minimum baseband input voltage, in voltage units
- Vmax = maximum baseband input voltage, in voltage units
- N = number of quantized output levels
- OutState = output is an integer representing quantization level = nonzero; output is scaled same as input but is quantized = zero
- Rout = output resistance, in ohms

Notes/Equations

1. This quantizer model outputs a delayed, uniformly quantized representation of the baseband portion of the input voltage. The input impedance is infinite. The delay is fixed at one timestep. The quantization state output mode is in offset binary format.

   The input quantization level is equal to \( \Delta = (V_{\text{max}} - V_{\text{min}}) / (N - 1) \). Any input less than \( (V_{\text{min}} + \Delta / 2) \) is assigned to state 0. Any input greater than \( (V_{\text{max}} - \Delta / 2) \) has a state equal to \( (N - 1) \). If the OutState parameter is true, this quantization state is directly output. If the OutState parameter is equal to zero, the quantization state is scaled and offset, \( V_{\text{out}} = Q/\Delta + V_{\text{min}} \), to match the input voltage.

2. This model works in transient and circuit envelope simulation.

3. An example of using this component along with SampleHoldSML follows.
ResetSwitch (Reset Switch)

Symbol

Parameters
None

Notes/Equations
1. This reset switch is an ideal switch that is closed at time=0 and open for time>0. Its main use is in time domain simulation modes (transient and circuit envelope) to allow the resetting of components such as ideal integrators. Often these blocks that have infinite gain at dc do not allow the circuit simulator to converge, but by keeping them reset at time=0, a valid initial solution can be obtained prior to the actual time domain simulation and then the switch can be opened to observe the time domain response. In steady-state analysis, such as dc or harmonic balance, time is always 0 so this switch remains closed.
SampleHoldSML (Sample Hold)

Symbol

![Symbol Diagram]

Parameters

\( F_{\text{nom}} = \) nominal input and output frequency, in hertz

Notes/Equations

1. This model samples the input signal and holds it until the next sample event. It is a tuned model; it selects and holds just the signal at the input harmonic closest to the frequency specified by \( F_{\text{nom}} \). If there is no analysis harmonic frequency close enough to the \( F_{\text{nom}} \) frequency, a warning is issued and the output is 0. The input impedance of both the sample clock input and the signal input is infinite. The output is a voltage source with a fixed impedance of 0.1 ohm.

2. The sampling instant is defined to be when the sample clock input rises through the 0.5V fixed threshold. If the baseband mode of operation is specified (\( F_{\text{nom}} \sim 0 \text{ Hz} \)), this behaves as a normal sample and hold; the baseband input voltage at the sampling instant is output and then held indefinitely at this constant value until the next sample. If an envelope carrier frequency is specified by \( F_{\text{nom}} \), the signal component at this envelope harmonic frequency is selected and its complex value at the sampling instant is held at the output. This held output appears at the same envelope harmonic frequency. For example, if an envelope analysis with a fundamental of 1 GHz is performed with a timestep of 0.1nsec and an offset frequency source at 1.001 GHz is sampled, the complex value of the envelope is sampled and held constant. The output will appear to be at the 1 GHz fundamental frequency since the complex envelope is no longer changing, until the next sample event.

3. The Sampler model samples the entire spectrum and outputs just a real baseband voltage.

4. SampleHoldSML works in transient and circuit envelope simulation. In transient simulation, \( F_{\text{nom}} \) has no effect because all signals are baseband only.
Switch and Algorithmic Components

**Sampler (Sampler)**

**Symbol**

![Symbol Image]

**Parameters**

- **Ton** = ON-state pulse width, switch in low impedance state, in seconds
- **Ron** = ON-state resistance, switch in low impedance state, in ohms
- **S11** = input reflection coefficient
- **Z0** = input reference impedance, in ohms

**Notes/Equations**

1. Sampler is a linear behavioral model of a high-frequency sampler. It can also be used to model the sampling efficiency and droop of lower frequency sample-holds.

2. This model works in transient and circuit envelope simulation. In envelope simulation, the input signal is determined by transforming all the spectral input voltages at the sampling instant to determine the total, real instantaneous voltage. Prior to being sampled, the input signal is first filtered with an ideal sinc() filter determined by the Ton pulse width. The input match can be complex, but since it needs to represent a causal response it cannot be complex at dc.

3. The input impedance looking into the sampler is determined by Z0 and S11 with the standard formula:

   \[ Z_{in} = \frac{Z_0(S11 + 1)}{S11 - 1} \]

   Usually, S11=0 and Z0=50 results in the input impedance of the sampler being 50 ohms.

4. The Clock/LO input impedance is infinite, and the sampling instant is defined only by the baseband portion of the input signal and occurs whenever the baseband signal passes through 0.5V with a positive slope.
5. Sampler has two basic modes of operation. If Ron is equal to 0, then the output is an ideal voltage source with a value equal to the last sampled value of the filtered input voltage; if Ron is not equal to 0, then the output impedance is time-varying. In the quiescent state, it is 1 Ohm. However, when a sample occurs, the impedance for that time point is reduced to a sampler resistance equal to Ron \times \text{timestep}/\text{Ton}. The actual sampler efficiency is then determined by sampler Ron and Ton values, and the load capacitance on the sampler output. If Ton \gg Ron \times CLoad, the sampler will behave with 100% efficiency. In a microwave sampler, this efficiency is typically much less than 100%. The hold time constant in this sampler mode is determined strictly by the capacitive and resistive load placed on the sampler output by the user.

The Ron sampler parameter should include all charging impedances, including the sampler switch impedance as well as the effect of any source impedance.

The output signal in Envelope mode is a baseband-only signal. No RF leakage is included in this model.

Note that due to the impulse nature of this sampler model output, the analysis integration order should be set to 1 (Backward Euler) when using this model. This is especially true when sampling rapidly changing input signals and when the sampler parameters are set for high sampling efficiency.

6. In the circuit envelope example in Figure 6-2, the input signal is a high-frequency sawtooth waveform with 15 harmonics. Its frequency is set 10 kHz above 1 GHz. The sampler is being driven at a 500 kHz rate so the output signal should be a replicated sawtooth at 10 kHz. Simulation results for two different combinations of Ron and Ton are shown in Figure 6-3. The high-efficiency mode tracks the input. The low-efficiency mode (50 ohm, 50 psec) shows its lowpass filtering impact. Note the two different time scales: 2 nsec for the input trace and 200 \mu sec for the output traces.

The example in Figure 6-4 shows a low-frequency application that has a significant droop due to the finite output resistance. Simulation results are shown in Figure 6-5.
Switch and Algorithmic Components

Figure 6-2. Circuit Envelope Example, High-Frequency Application

Figure 6-3. Simulation Results
Figure 6-4. Circuit Envelope Example, Low-Frequency Application

Figure 6-5. Simulation Results
SerialParallel (Serial to Parallel Shift Register)

Symbol

Parameters

InputRate = serial input data clock rate, in hertz
LSB_First = serial data arrives with least significant bit first = YES; NO = most significant bit arrives first
Delay = initial synchronization delay, in seconds
OutputBits = number of bits in output word
IntegerOut = scale output data as integers (instead of from −1 to 1): YES, NO

Notes/Equations

1. This SerialParallel shift register model is used to convert a serial data stream into a multiple bit word. The fixed serial input bit rate is specified by the InputRate parameter, with an initial synchronization delay specified by the Delay parameter. The serial data can be interpreted as either LSB or MSB first, depending on the state of the LSB_First parameter. The input impedance is infinite. A voltage less than 0.5V is a logical zero.

2. The number of bits in each output word is specified by OutputBits. The output is updated at a rate equal to InputRate/OutputBits. The open circuit output voltage is an integer value from 0 to $2^{\text{OutputBits}}-1$, if the IntegerOut parameter is true. Otherwise, the output is scaled from a −1 to +1V, with all 0s corresponding to −1V and all 1s to +1Vs. The output impedance is fixed at 0.1 ohm.

3. This model works in transient and circuit envelope simulation. Only the baseband portion of circuit envelope voltage is used.
SPDT_Dynamic (Single Pole Double Throw Switch, Dynamic)

Symbol

Parameters
Ron = on-state resistance of switch, in ohms
Roff = off-state resistance of switch, in ohms

Range of Usage

Pin 1 is the input, pins 2 and 3 are output, pin 4 is the control voltage, Vc.
If Vc > 2V, R1 = Ron, R2 = Roff
If Vc < 1V, R1 = Roff, R2 = Ron
When Vc increases from 1V to 1.5V, R2 changes from Ron to Roff
When Vc increases from 1.5V to 2V, R2 = Roff and R1 changes from Roff to Ron
At Vc = 1.5V, R1 = R2 = Roff

Notes/Equations
1. This SPDT_Dynamic switch model, as opposed to the SPDT_Static model, can be used to dynamically switch states in response to the input control voltage. The input impedance of the control voltage port is infinite. Whenever the control voltage is greater than 2.0V, the input is connected to the (c>2) output with a resistance equal to Ron; otherwise, there is a resistance of Roff ohms between the two pins. Similarly, whenever the control voltage is less than 1.0V, the input is connected to the (c<1) output with a resistance of Ron; otherwise, there is a resistance of Roff between the two pins. Note that when the control
Switch and Algorithmic Components

- voltage is between 1.0V and 2.0V, the switch is open with respect to both outputs, simulating the break operation before making connection.

2. Capacitances, leakage currents, or other non-idealities are not included in this model.

3. This model is primarily meant for usage in the time domain (transient and circuit envelope) simulation with baseband control voltages. Due to the instantaneous, abrupt nature of the switching action, driving the control input with a carrier frequency input, in either harmonic balance or circuit envelope, should be avoided or done with careful consideration of the number of harmonics required and potential convergence problems.
SPDT_Static (Single Pole Double Throw Switch, Static)

Symbol

Parameters
State = state of switch: for 0, nodes 1 and 2 are connected; for 1, nodes 1 and 3 are connected
F1 = first frequency breakpoint
F2 = second frequency breakpoint
F3 = third frequency breakpoint
Loss1 = attenuation for frequencies \(\leq F1\), in dB
Loss2 = attenuation for frequencies \(\gt F1 \leq F2\), in dB
Loss3 = attenuation for frequencies \(\gt F2 \leq F3\), in dB
VSWR1 = VSWR at both ports for frequencies \(\leq F1\)
VSWR2 = VSWR at both ports for frequencies \(\gt F1 \leq F2\)
VSWR3 = VSWR at both ports for frequencies \(\gt F2 \leq F3\)
Isolat = isolation, in dB
ZRef = reference impedance for all ports, in ohms
Temp = temperature, in degree C
CheckPassivity = check passivity flag: if set to yes (default), a passivity check is performed and a warning is output if the device is not passive; if set to no, a passivity check is not performed.

Notes/Equations
1. This model is based on S-parameters (\(S_{12} = S_{21}, S_{13} = S_{31}\)).
2. Up to three frequency break points can be used to define four bands, with different losses and VSWR for each frequency band.
   Enter loss as a positive attenuation. In other words, a 0.5 dB loss is entered as \(\text{Loss} = 0.5\), not \(\text{Loss} = -0.5\).
Switch and Algorithmic Components

3. Isolation is assumed to be constant across all frequency bands. When using this device in a transient simulation, the switch setting must be constant.
SwitchV (Voltage Controlled Switch)

Symbol

Parameters
Model = name of a SwitchV_Model
R1 = resistance at voltage 1, in ohms
V1 = voltage 1, in volts
R2 = resistance at voltage 2, in ohms
V2 = voltage 2, in volts

Range of Usage
R1, R2 > 0
V1 \neq V2

Notes/Equations
1. This component implements a voltage controlled switch. The switch resistance between nodes 1 and 2 varies as a function of the applied control voltage (V_{CON} is the voltage between nodes 3 and 4):

\[
R_{SW} = \begin{cases} 
R1 & \text{if } V_{CON} \leq V1 \\
\text{f}(V_{CON}) & \text{if } V1 \leq V_{CON} \leq V2 \\
R2 & \text{if } V_{CON} \geq V2 
\end{cases}
\]

where
Switch and Algorithmic Components

For these equations, it is assumed that \( V_1 < V_2 \). If \( V_2 < V_1 \), swap \( R_1 \) and \( V_1 \) with \( R_2 \) and \( V_2 \) in the equation above.

\( V_{CON} \) is defined so that \( f(V_1) = R_1 \) and \( f(V_2) = R_2 \) and provides a smooth transition between states. This is a plot of the switch resistance as a function of the control voltage for \( V_1 = 1 \), \( R_1 = 10 \), \( V_2 = 3 \), \( R_2 = 10000 \).

2. The Model name is optional. If the model name is not specified, then \( R_1 \), \( V_1 \), \( R_2 \) and \( V_2 \) must all be specified on the SwitchV instance; if the model name is specified, then \( R_1 \), \( V_1 \), \( R_2 \) and \( V_2 \) on the SwitchV instance are optional and override the values specified in the SwitchV_Model.

3. \( R_1 \) and \( R_2 \) are model parameters, not the resistance at nodes. Convergence problems may occur if the difference between \( R_1 \) and \( R_2 \) is too large.

4. \( V_1 \) and \( V_2 \) are model parameters, not node voltages. Convergence problems may occur if the difference between \( V_1 \) and \( V_2 \) is too small.

5. This component works in all analyses, including transient and circuit envelope simulation. If a SwitchV_Model item is not present, the SwitchV Model parameter should be blank.

6. Table 6-2 lists the DC operating point parameters that can be sent to the dataset.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is</td>
<td>Current</td>
<td>A</td>
</tr>
<tr>
<td>Power</td>
<td>DC power dissipated</td>
<td>W</td>
</tr>
<tr>
<td>Vs</td>
<td>Voltage</td>
<td>V</td>
</tr>
</tbody>
</table>
SwitchV_Model (Voltage Controlled Switch Model)

Symbol

Parameters
R1 = resistance at voltage 1, in ohms
V1 = voltage 1, in volts
R2 = resistance at voltage 2, in ohms
V2 = voltage 2, in volts
AllParams = DataAccessComponent-based parameters

Range of Usage
R1, R2 > 0
V1 ≠ V2

Notes/Equations
1. This model supplies values for a SwitchV component.
2. The range between R1 and R2 should not be too large, or convergence problems may occur.
3. The difference between V1 and V2 should not be too small, or convergence problems may occur.
4. R1, V1, R2 and V2 can be optionally specified on the SwitchV instance that uses this model; those values override the model values.
5. This model works in transient and circuit envelope simulation.
6. Use AllParams with a DataAccessComponent to specify file-based parameters (refer to the DataAccessComponent). Note that model parameters that are explicitly specified take precedence over those specified via AllParams.
VSum (Voltage Summer)

Symbol

Parameters
None

Notes/Equations
1. This component has infinite input impedance and zero output impedance. Output voltage is equal to voltage summed from the two input ports.
Chapter 7: Tx/Rx Subsystems
Tx/Rx Subsystems

RF_PA_CKT (RF Power Amplifier Circuit)

Symbol

Parameters

None

Notes/Equations

1. This is a 2-stage BJT amplifier with the center frequency of 2GHz, small signal gain of ~30dB, and maximum power output of ~15dBm. This circuit is provided for the convenience of easy inclusion of a circuit level amplifier in an Agilent Ptolemy Envelope cosimulation.
RF_RX_SML (RF Receiver)

Symbol

Parameters
RX_AntTemp = Receiving antenna noise temperature, in Kelvin (default: 150)
RX_Gain = Receiver gain, in dB (default: 80 dB)
RX_NF = Receiver noise figure in dB (default: 5 dB)
RF_Freq = RF frequency, in MHz, GHz, or THz (default: 900 MHz)
RF_BW = RF bandwidth, in Hz, kHz, MHz, or GHz (default: 25 MHz)
IF_Freq1 = First IF frequency in Hz, kHz, MHz, or GHz (default: 100 MHz)
IF_Freq2 = Second IF frequency in Hz, kHz, MHz, or GHz (default: 400 KHz)
IF_BW = IF bandwidth in Hz, kHz, MHz, or GHz (default: 30 KHz)
IP3in = Receiver input IP3, in pW, nW, uW, mW, W, kW, dBm (default: -25 dBm)
RIn = Input resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)
ROut = Output resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

Notes/Equations
1. This is a super-heterodyne RF receiver subsystem. Input is an RF signal; output is an IF signal. Three CW spurs are combined with the input RF signal to simulate the receiver interference performance.

This subnetwork is provided for easy inclusion of an RF receiver in an Agilent Ptolemy Envelope co-simulation.
Tx/Rx Subsystems

**RF_TX_SML (RF Transmitter)**

**Symbol**

![Symbol Diagram]

**Parameters**

- **IF_Freq** = IF frequency in Hz, kHz, MHz, or GHz (default: 400 kHz)
- **RF_Freq** = RF frequency, in Hz, kHz, MHz, or GHz (default: 900 MHz)
- **RF_BW** = RF bandwidth, in Hz, kHz, MHz, or GHz (default: 30 KHz)
- **TX_Gain** = Transmitter gain, in dB (default: 80 dB)
- **PSat** = Saturated power at output, in dBm, dBW, nW, uW, mW, W, or kW (default: 35 dBm)
- **RIn** = Input resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)
- **ROut** = Output resistance in Ohm, mOhm, kOhm, MOhm, GOhm (default: 50 Ohm)

**Notes/Equations**

1. This is an RF transmitter subsystem. Input is an IF signal; output is an RF signal.
   
   This subnetwork is provided for easy inclusion of an RF transmitter in an Agilent Ptolemy Envelope co-simulation.
Chapter 8: System Data Models

Introduction

The System - Data Models palette contains two types of components: those which enable the simulation-based extraction of behavioral profiles from circuit models and those which perform behavioral modeling using these profiles. There is an inherent pairing of these two types of components because parameters used to define behavioral profiles in data-based representation are specific to modeling theories and therefore data model requirements. Data-based behavioral modeling is an important tool for the bottom-up verification process; thus, each extractor-data model pair of this palette is also known as a verification model extraction (VME) pair. The most common usage of a VME pair during RFIC and MMIC design flows is as follows:

- Creation of a circuit level design with circuit level components and/or parametric behavioral components during the top-down design phase. Parameter-based behavioral models are located in the Filters, System - Amps & Mixers, and System - Mod/ Demod palettes. These models are characterized by a few independent parameters such as frequency, power and load.

- Creation of a Harmonic Balance-based dataset or data file profiles of these circuit level models using the VME extractor components from the System - Data Models palette. Use the VME data model counterparts to recreate the behavior of the circuit level models. Compare circuit- and data-based behavioral models; calibrate the data-based behavioral models prior to hardware prototyping the circuit.

- Using measurement instruments to extract behavioral datasets or data files from hardware prototypes from the foundry. Using the calibrated VME data model to generate a hardware behavioral profile. Compare the hardware behavioral profile with the simulation-based behavioral profile to estimate the success of prototyping.

The parameter-based behavioral models typically provide superior speed whereas data-based behavioral models typically provide superior accuracy. The differences between parameter- and data-based behavioral models justify the palette emphasis on ADS flow (separate palettes for parametric and data-based models) rather than functionality (separate palettes for amplifiers, mixers, modulators each containing both parametric and data-based models).
System Data Models

The BehavioralModels example suite was developed to promote the usability of the System-Data Models components. This family of example projects demonstrates the use and characteristics of most VME models. To access the example project from the ADS Main window click on the following sequence of menu options: File > Example Project > BehavioralModels. Notes for each example project are available in the accompanying notes in ADS Documentation > Examples > Behavioral Models manual.

In addition to the data-based behavioral model components, the System-Data Models palette includes Balun3Port and Balun4Port; these components are also accessible from the System-Passive palette.

Classification of ADS System Data Models

ADS provides data-based modeling for system-level VME components of the following categories:

- Amplifiers: AmplifierH1H2, AmplifierP2D, AmplifierS2D, AmpLoadPull, VCA_Data
- Mixers: MixerHBdata, MixerIMT2
- Modulators / Demodulators: IQ_Mod, IQ_Demod

Most data models accept customized dataset (*.ds) or data file (*.p2d, *s2d) inputs. These inputs are typically provided by data model extractors. Identified by the suffix Setup or setup, these extractors are prefabricated subcircuits that run a predetermined simulation with user-settatable sweep values on the circuit level model of interest and report the input and output stimuli in the dataset or data file form. The user is encouraged to push into the extraction subcircuits and be familiar with the measurement metrics that are being reported to the dataset or data file.

Most system data models have their own data extractors; however, there are exceptions (such as AmplifierS2D and MixerIMT2 data models) for which data files must be manually generated or retrieved from a measurement instrument.

Table 8-1 summarizes the availability and scope of each VME model currently available in ADS. This chapter describes component parameters and usage in further detail. For a discussion of the use of the ADS data model components from an application point of view please refer to the series of Application Notes being released under the title Behavioral Modeling in ADS. The first of these notes which targets the use of the VME pair: AmplifierP2D_Setup and data model AmplifierP2D is being released in ADS 2003C.
### Table 8-1. Data-Based Behavioral Modeling Components and Example Projects in ADS.

<table>
<thead>
<tr>
<th>Extractor Model</th>
<th>Data Model</th>
<th>Data Type</th>
<th>Data Type</th>
<th>Comments on Scope of Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmpH1H2_Setup</td>
<td>AmpH1H2</td>
<td>.ds</td>
<td></td>
<td>Models nonlinearity at odd and even harmonics using circuit-based data at fundamental and second harmonics. Refer to Examples &gt; BehavioralModels &gt; AmpH1H2_prj</td>
</tr>
<tr>
<td>AmplifierP2D_Setup</td>
<td>AmplifierP2D</td>
<td>.p2d</td>
<td></td>
<td>Models nonlinearity and noise at fundamental frequency using circuit-based data at fundamental frequency. For circuit envelope simulations, modeling nonlinear distortion at fundamental frequency due to interference from spectral components within the envelope band. Refer to Examples &gt; BehavioralModels &gt; AmplifierS2DandP2D_prj</td>
</tr>
<tr>
<td>(Manual extraction)</td>
<td>AmplifierS2D</td>
<td>.s2d</td>
<td></td>
<td>Models nonlinearity and noise at fundamental and odd-order harmonic frequencies using circuit-based data at fundamental frequency. Assumes odd-order polynomial fitting for harmonic calculations. Refer to Examples &gt; BehavioralModels &gt; AmplifierS2DandP2D_prj</td>
</tr>
<tr>
<td>LoadPullSetup</td>
<td>AmpLoadPull</td>
<td>.ds</td>
<td></td>
<td>Models nonlinearity at fundamental frequency for various input and output impedance conditions using circuit-based data at single frequency for a range of source and load conditions. Refer to Examples &gt; BehavioralModels &gt; AmpLoadPull_prj</td>
</tr>
<tr>
<td>VCA_Setup</td>
<td>VCA_Data</td>
<td>.ds</td>
<td></td>
<td>Models nonlinearity at fundamental frequency for various settings of control voltage using circuit-based data at single frequency for a range of bias conditions. Refer to Examples &gt; BehavioralModels &gt; VCA_prj</td>
</tr>
<tr>
<td>MixerHBsetup</td>
<td>MixerHBdata</td>
<td>.ds</td>
<td></td>
<td>Models nonlinearity at fundamental, mixing product frequencies at the mixer output (IF) due to 1-tone input signals (RF and LO) based on 1-tone harmonic balance characterization of circuit level mixer. Refer to Examples &gt; BehavioralModels &gt; MixerHBdata_prj</td>
</tr>
<tr>
<td>(Manual extraction)</td>
<td>MixerIMT</td>
<td>.imt</td>
<td></td>
<td>Models nonlinearity at fundamentals and mixing product frequencies at the mixer output (IF) based on 1-tone input signals and a table of mixing coefficients. Refer to Examples &gt; BehavioralModels &gt; MixerIMT_prj</td>
</tr>
<tr>
<td>(Manual extraction)</td>
<td>MixerIMT2</td>
<td>.imt</td>
<td></td>
<td>Models nonlinearity at fundamentals and mixing product frequencies at the mixer output (IF) based on 1-tone input signals and a table of mixing coefficients. Refer to Examples &gt; BehavioralModels &gt; MixerIMT2_prj</td>
</tr>
<tr>
<td>IQ_Mod_Setup</td>
<td>IQ_Mod_Data</td>
<td>.ds</td>
<td></td>
<td>Models large-signal in-phase and quadrature components during modulation. Refer to Examples &gt; BehavioralModels &gt; IQ_Mod_prj</td>
</tr>
<tr>
<td>IQ_Demod_Setup</td>
<td>IQ_Demod_Data</td>
<td>.ds</td>
<td></td>
<td>Models large-signal in-phase and quadrature components during demodulation. Refer to Examples &gt; BehavioralModels &gt; IQ_Demod_prj</td>
</tr>
</tbody>
</table>

† Manual generation of an S2D file based on 1-tone harmonic balance simulations is shown in design CKT_S2D_HB_1tone.dsn and data display schematic Create_Motorola_PA_S2DFile.dsn of example project AmplifierS2DandP2D_prj.

†† This is an obsolete component.
System Data Models

AmpH1H2 (Amplifier/Fundamental and 2nd Harmonic vs. Input Power)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Name of dataset containing data for this amplifier model.</td>
<td></td>
<td>&quot;dataset.ds&quot;</td>
</tr>
<tr>
<td>G1expr</td>
<td>Gain expression for fundamental frequency.</td>
<td></td>
<td>&quot;Vout[1]/Vin[1]&quot;</td>
</tr>
<tr>
<td>G2expr</td>
<td>Gain expression for second harmonic frequency.</td>
<td></td>
<td>&quot;Vout[2]/Vin[2]&quot;</td>
</tr>
<tr>
<td>SP11</td>
<td>Forward reflection coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP22</td>
<td>Reverse reflection coefficient.</td>
<td></td>
<td>polar(0,180)</td>
</tr>
<tr>
<td>SP12</td>
<td>Reverse transmission coefficient.</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>NF</td>
<td>Noise figure.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>NFmin</td>
<td>Minimum noise figure at Sopt.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Sopt</td>
<td>Optimum source reflection for minimum noise figure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rn</td>
<td>Equivalent noise resistance.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>Z1</td>
<td>Reference impedance for Port1.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>Z2</td>
<td>Reference impedance for Port2.</td>
<td>Ohm</td>
<td></td>
</tr>
</tbody>
</table>

† The gain parameters must be updated to appropriate gain expressions based on the MeasEqn expressions of AmpH1H2_Setup component which was used to create the dataset. Refer to example project AmpH1H2_prj.
†† These parameters can be reported in any of the following complex number formats: x + j*y, polar(x,y), dbpolar(x,y), vswrpolar(x,y)

Notes/Equations

1. AmpH1H2 is a data-based system model of a circuit-level amplifier. The circuit-level amplifier is characterized by a dataset generated by the extractor component AmpH1H2_Setup. Various examples of the use of AmpH1H2 are provided in the example project AmpH1H2_prj.

2. This amplifier model does not require the explicit specification of nominal frequency at which non-linear modeling is to be performed. Although technically it can be used at any frequency during simulation, the dataset file contains the nonlinear profile at the single frequency point defined on the
AmpH1H2_Setup component during extraction. Hence, effectively the AmpH1H2 model is confined to reproducing behavior accurately at that frequency only. Since the AmpH1H2 model does not have a built-in warning mechanism to detect discrepancies between simulation and extraction frequencies, the user of this model needs explicit a priori knowledge of the extraction frequency during behavioral simulation.

3. AmpH1H2 models both odd- and even-order harmonics at the amplifier output. However, its reliance on the fundamental and second-order harmonic expressions $G_{1\text{expr}}$ and $G_{2\text{expr}}$, respectively, guarantee high accuracy only for these two spectral components. For details of the modeling accuracy of various harmonics in a 1-tone swept power Harmonic Balance simulation, refer to the data display plot behavioral_level_amp_SOITOI.dds in the example project AmpH1H2_prj.

4. The dataset contains information about the forward transmission characteristic $S_{21}$. The other $S$-parameters $S_{11}$, $S_{12}$ and $S_{22}$, can be set explicitly on the data model instance. Likewise, noise parameters $NF$, $NF_{\text{min}}$, $S_{\text{opt}}$ and $R_{n}$, can be set explicitly using relevant parameters on the model instance.

5. It is important to note that although port impedances $Z_1$ and $Z_2$ can be set to arbitrary values during behavioral simulation, the dataset is typically extracted at port impedance values of 50 Ohms. Ideally, the settings for $Z_1$ and $Z_2$ inside the subcircuit of the extractor model AmpH1H2_Setup should be set up as complex conjugates of the input and output impedances, respectively, of the circuit amplifier. The resulting dataset would then contain the behavior under matched source and load conditions. In order to translate the dataset into the behavior of AmpH1H2, the AmpH1H2 $Z_1$ and $Z_2$ parameters must then be set to the complex conjugates of AmpH1H2_Setup component's $Z_1$ and $Z_2$ values, i.e. to the original values of the circuit amplifiers' port impedances.
AmpH1H2_Setup (Amplifier/Fundamental and 2nd Harmonic vs. Input Power Setup)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Fundamental frequency. Must be a positive real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance analysis. Must be an integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Pin_Start</td>
<td>Start value of input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>-50 dBm</td>
</tr>
<tr>
<td>Pin_Stop</td>
<td>Stop value for input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>20 dBm</td>
</tr>
<tr>
<td>Pin_Step</td>
<td>Interval for input power sweep during harmonic balance analysis.</td>
<td>dB</td>
<td>10 dB</td>
</tr>
</tbody>
</table>

Notes/Equations

1. AmpH1H2_Setup performs a swept input power Harmonic Balance simulation of a circuit-level amplifier and generates a dataset for subsequent use by the data-based system model AmpH1H2. The AmpH1H2_Setup extractor model is demonstrated in the example project AmpH1H2_prj.

2. AmpH1H2_Setup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.

3. This component is capable of extracting circuit behavior at only one nominal frequency Freq. In order to characterize an amplifier over a range of frequencies, separate extractions must be performed and separate datasets generated at each frequency of interest.

4. In order for AmpH1H2 to produce an accurate model of the circuit-level amplifier characterized via AmpH1H2_Setup, the Order parameter for AmpH1H2_Setup must be large enough to prevent aliasing of higher-order frequency components. The recommended value is 5 for mildly non-linear circuits and 11-15 for highly non-linear circuits.
5. The linear input power sweep is defined by the parameters Pin_Start, Pin_Stop and Pin_Step are used directly by the underlying Harmonic Balance controller.

6. The extracted ADS dataset is assigned the name of the extraction design by default.
AmplifierP2D (P2D File Amplifier; FDD-Based, for Single Carrier Signal)

Symbol

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Freq</td>
</tr>
<tr>
<td>P2DFile</td>
</tr>
<tr>
<td>FilteringOption†</td>
</tr>
<tr>
<td>CEFreqSpacing†</td>
</tr>
</tbody>
</table>

† The FilteringOption and CEFreqSpacing parameters are provided (beginning with 2003C) to enhance circuit envelope simulation capabilities; refer to note 8 and note 10 for details. The default values suppress effects of FilteringOption and CEFreqSpacing parameters to ensure backwards compatibility.

Notes/Equations

1. AmplifierP2D is a data-based system model of a circuit-level amplifier. The circuit-level amplifier is characterized by a P2D file generated either by the extractor component AmplifierP2D_Setup in a simulation environment or by a measurement instrument such as a network analyzer. Various examples of the use of the AmplifierP2D data model are provided in the example project AmplifierS2DandP2D_prj.

2. A P2D file, named *.p2d, contains small- and large-signal 2-port S-parameter data with optional noise parameters and intermodulation table data. The AmplifierP2D model uses data related to amplifier modeling only; it ignores frequency translation and intermodulation table data that may be contained in a generic P2D file. The referenced P2D file should reside in the data subdirectory of the current project. For details on P2D file format, refer to the “P2D Format” section in the Circuit Simulation manual.

3. The AmplifierP2D model blocks DC. For CE simulations, baseband signals are blocked.

4. For small-signal simulations such as AC and S-parameter analyses, AmplifierP2D uses the small-signal S-parameters of the P2D file only. For
S-parameter analysis, the nominal frequency must be set as Freq=\texttt{freq}, where \texttt{freq} is the ambient simulation frequency. If the Freq parameter value falls outside the range of frequencies in the small-signal section of the P2D file, an error message is reported in the simulator log window and the simulation terminated. Linear interpolation of S-parameters is performed within the small-signal range to emulate behavior at a frequency not explicitly registered in the data file. Linear extrapolation is enabled but not recommended outside the small-signal frequency range because modeling accuracy cannot be guaranteed outside the data points of the P2D file.

5. For large-signal frequency analysis such as those based on Harmonic Balance, the power-dependent S-parameters from the ACDATA block are used in conjunction with small-signal parameters. For these simulations, Freq must be set to the fundamental frequency explicitly or through the pre-defined variable \texttt{_freq1}. Setting Freq=\texttt{freq} will lead to errors. For an N-tone Harmonic Balance simulation any of N frequencies specified on the controller may be chosen as nominal frequency for the AmplifierP2D component by setting Freq=\texttt{_freqX}, where X is in the integer range \([1, N]\).

6. If the Freq parameter value falls outside the range of frequencies in the large-signal section of the P2D file, a warning message is reported in the simulator log window although the simulation is allowed to proceed. Linear interpolation along the frequency axis is permitted within the small- and large-signal range but extrapolation is not advised on either side of the large-signal limits because of insufficient data. The user is urged to scan the simulator log window at the end of any large-signal simulation involving the AmplifierP2D component to ensure that frequency limits were not exceeded. This frequency range check feature is highlighted in the example design BEH\_P2D\_Freq\_rangechecking.dsn.

7. If the power incident at the input of the amplifier model during large-signal frequency domain analyses, such as Harmonic Balance, LSSP or Circuit Envelope simulations, exceeds the maximum value of \(P1\) in the P2D file then a warning message is issued to the simulator log window about unsupported modeling for such high drive levels.

8. The FilteringOption parameter allows the user to regulate modeling of distortion effects at the fundamental frequency due to other frequencies within the envelope bandwidth during a circuit envelope simulation. Some harmonics and intermods produced by the non-fundamental tones within the CE-band contribute to the distortion of the fundamental tone at the output of the amplifier for high drive levels at the input. In the design of complex circuit level
amplifiers this distortion is either regulated at the amplifier input or output using a narrowband filter. In order to imitate this feature during behavioral simulation AmplifierP2D enables the user to set the FilteringOption parameter to PreFilter or PostFilter so that appropriate filtering effects can be modeled without changing the P2D data file. It is useful to perform such CE-band distortion evaluation for certain applications (communications systems, for example). The NoFilter default setting suppresses estimation of envelope bandwidth distortion at the nominal frequency Freq; this setting is sufficient for all non-CE simulations.

9. The PreFilter or PostFilter option is for use during circuit envelope simulations only. Prefiltering of non-fundamental frequencies reduces the potential of distortion close to power saturation of the fundamental frequency at the output; postfiltering allows maximal non-linear distortion to occur at amplifier output. Maximally flat filtering is assumed across the entire envelope bandwidth. This distortion modeling feature is highlighted in the example design BEH_P2D_CE_filter.dsn.

10. The CEFreqSpacing parameter is used to select the granularity of envelope band frequencies that are allowed to impact the distortion of the fundamental frequency at amplifier output. There is an inherent performance trade-off in using this parameter: the smaller its value the more accurate the simulation but the longer the duration of the CE simulation. If filtering is enabled and CEFreqSpacing exceeds half the value of Freq a warning message is reported to the simulation log window and simulation proceeds without the use of filtering due to violation of the Nyquist sampling requirement for CEFreqSpacing. The recommended value for CEFreqSpacing is \(10^{-3}\times(Freq)\). This feature is highlighted in the example design BEH_P2D_CE_sample.dsn.

11. AmplifierP2D is implemented using the FDD model. Unlike the SML models Amplifier and Amplifier2 or the VME model AmplifierS2D, AmplifierP2D does not produce harmonics or intermods in a Harmonic Balance simulation.

In a Harmonic Balance analysis, the only frequency component that can pass through AmplifierP2D is that specified by Freq. Signals at all other frequencies will see a ground at the input of AmplifierP2D. However, in a Circuit Envelope simulation, the envelope signal around the carrier frequency Freq will pass through AmplifierP2D and cause distortion at the fundamental Freq, whereas all signals outside the envelope band will be ignored.

To simulate multi-tone behavior involving harmonic modeling, consider using the AmpH1H2 or AmplifierS2D models.
AmplifierP2D_Setup (P2D File Amplifier; FDD-Based, for Single Carrier Signal Setup)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fname</td>
<td>Name of P2D file to be generated by extractor</td>
<td></td>
<td><em>p2dfile.p2d</em></td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance simulation inside extractor.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Freq_Start</td>
<td>Start value for sweep of large-signal frequencies.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Freq_Stop</td>
<td>Stop value for sweep of large-signal frequencies.</td>
<td>Hz</td>
<td>2.5 GHz</td>
</tr>
<tr>
<td>Freq_Step</td>
<td>Interval for linear sweep of large-signal frequencies.</td>
<td>Hz</td>
<td>0.5 GHz</td>
</tr>
<tr>
<td>Pin_Start</td>
<td>Start value for sweep of large-signal input power.</td>
<td>dBm</td>
<td>-10 _dBm</td>
</tr>
<tr>
<td>Pin_Stop</td>
<td>Stop value for sweep of large-signal input power.</td>
<td>dBm</td>
<td>15 _dBm</td>
</tr>
<tr>
<td>Pin_Step</td>
<td>Interval for linear sweep of large-signal input power.</td>
<td>dB</td>
<td>5 _dB</td>
</tr>
<tr>
<td>SSFreq_Start</td>
<td>Start value for sweep of small-signal frequencies.</td>
<td>Hz</td>
<td>5 _dB</td>
</tr>
<tr>
<td>SSFreq_Stop</td>
<td>Stop value for sweep of small-signal frequencies.</td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>SSFreq_Step</td>
<td>Interval for linear sweep of small-signal frequencies.</td>
<td>Hz</td>
<td></td>
</tr>
</tbody>
</table>

† Only linear sweeps using intervals of Freq_Step, Pin_Step, SSFreq_Step are permitted frequency and power extractions using the AmplifierP2D_extractor.
†† Small-signal frequency settings, if left unspecified, will default to large-signal frequency specifications. The default behavior of this component is to enforce SSFreq_Start <= Freq_Start, SSFreq_Stop >= Freq_Stop and SSFreq_Step <= Freq_Step. Violations of this will result in internal correction with warning messages being reported to the simulator log window. Refer to note 4 for modifying this default behavior.
Notes/Equations

1. AmplifierP2D is a data file extractor component used to create behavioral profiles of a circuit-level amplifier in the P2D format. The use of this component is highlighted in the design CKT_P2D_extraction.dsn of the example project AmplifierS2DandP2D_prj.

2. AmplifierP2D_Setup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.

3. AmplifierP2D_Setup performs a linear swept input frequency over linear swept input power P2D (Harmonic Balance) simulation of a circuit-level amplifier and generates a P2D file for subsequent use by the data-based system model AmplifierP2D. The Freq_ and Pin_ parameters regulate the large-signal frequency and input power sweeps, respectively.

4. The small-signal frequency sweep can be specified independently of the large-signal frequency sweep using SSFreq_ parameters. When unspecified, these values internally default to large-signal values. When specified in violation of the rule that SSFreq_Start <= Freq_S, SSFreq_Stop >= Freq_S and SSFreq_Step <= Freq_Step, the rule is imposed internally during simulation with warning messages registered in the simulator log window. This ensures that, at the very least, the small-signal frequency sweep spans the large-signal frequency sweep and that it is at least as finely grained. This is the coarsest granularity for reliable modeling using the AmplifierP2D data model.

5. In order for AmplifierP2D to produce an accurate model of the circuit-level amplifier characterized via AmplifierP2D_Setup, the Order parameter for AmplifierP2D_Setup must be large enough to prevent aliasing of higher-order frequency components. Because the AmplifierP2D data model reads and operates primarily on its assigned fundamental frequency, it is important to provide an accurate estimate of the fundamental response at the circuit level amplifier output. Therefore, it is important to use a large order of harmonics in AmplifierP2D_Setup to reconstruct the fundamental accurately during data modeling. The recommended value is 5 for mildly non-linear circuits and 11-15 for highly non-linear circuits.

6. For information regarding the P2D data format, refer to the “P2D Format” section in the Circuit Simulation manual.
## AmplifierS2D (S2D File Amplifier, Polynomial Model for Nonlinearity)

### Symbols:

![Symbol](image)

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2DFile</td>
<td>Filename for S2D data. Enumerated as (S2D filename, File Based) †</td>
<td></td>
<td>&quot;s2dfile.s2d&quot;</td>
</tr>
<tr>
<td>SSFreq</td>
<td>Small signal frequency for interpolating S-parameters. Enumerated as (auto, freq, _freq1, _freq2, _freq3).</td>
<td></td>
<td>auto††</td>
</tr>
<tr>
<td>InterpMode</td>
<td>Interpolation mode. Enumerated as [Linear, Cubic Spline, Cubic, Value Lookup, ...., Value]</td>
<td>Linear</td>
<td></td>
</tr>
<tr>
<td>InterpDom</td>
<td>Interpolation geometry. Enumerated as [Data Based, Rectangular, Polar, DB]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCFreq †††</td>
<td>Reference frequency for gain compression. Must be a positive real number if specified.</td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>VarName</td>
<td>Variable that parameterizes S2D data e.g. &quot;temp&quot; or &quot;bias&quot;. This string must be present in the S2D file with a numeric value assigned to it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VarValue</td>
<td>Value of parametric variable in S2D file.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpNoncausalLength ‡</td>
<td>Non-causal function impulse response order.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpMode ‡</td>
<td>Convolution mode.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpMaxFreq ‡</td>
<td>Maximum frequency to which device is evaluated. Must be a positive real number if specified.</td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>ImpDeltaFreq ‡</td>
<td>Sample spacing in frequency.</td>
<td>Hz</td>
<td></td>
</tr>
<tr>
<td>ImpMaxOrder ‡</td>
<td>Maximum allowed impulse response order. Must be a positive integer if specified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpWindow ‡</td>
<td>Smoothing window. Must be a positive integer if specified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpRelTol ‡</td>
<td>Relative impulse response truncation factor. Must be a positive real number if specified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImpAbsTol ‡</td>
<td>Absolute impulse response truncation factor. Must be a positive real number if specified.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† The "S2D Filename" option allows the entry of an S2D file directly. The "File Based" option enables the use of a DataAccessComponent (DAC) for reading in non-S2D file based data.

†† Implies ".freq1" for non-DC part of HB/CE/LSSP simulations. It implies "freq" for all other simulations. For multitone simulations setting SSFreq=._freq1 explicitly gives better results. In general ".freqX" allows the specification of any one of the X tones in an X-tone Harmonic Balance simulation, e.g. selecting SSFreq=._freq2 sets the small-signal frequency of interest to be the second tone of a 2+ tone HB simulation.

††† If manually specified, this frequency must explicitly exist in the S2D file. No interpolation or extrapolation is done along frequency axis for estimating compression characteristics.

‡ These parameters are also used as specifications for the Amplifier and Amplifier2 components.
System Data Models

Range of Usage

S2D gain compression valid parameter ranges:

- GCOMP3: IP3 > 1DBC + 10.6
- GCOMP4: IP3 > PS + 8.6
- GCOMP5: PS > 1DBC + 3
- GCOMP6: PS > 1DBC + 3, IP3 > 1DBC + 10.6

Notes/Equations

1. AmplifierS2D is a data-based system model of a circuit-level amplifier. The circuit-level amplifier is characterized by an S2D file generated manually from a single-tone Harmonic Balance simulation of a circuit model as shown in the design CKT_S2D_HB_1tone.dsn in example project AmplifierS2DandP2D_prj. Various examples of the use of AmplifierS2D are provided in this project.

2. An S2D file, named *.s2d, contains small-signal 2-port S-parameter data with optional noise parameters. In addition, it contains small-signal S-parameter data blocks, noise data blocks, and one of the possible seven gain compression blocks {GCOMP1, ..., GCOMP7}. For information regarding the small- or large-signal S-parameter *.s2d data format, refer to the “S2D Format” section in the Circuit Simulation manual. The AmplifierS2D model uses data related to amplifier modeling only; it ignores frequency translation data that may be contained in a generic S2D file. The referenced S2D file should reside in the data subdirectory of the current project.

3. The AmplifierS2D model blocks DC. For CE simulations, baseband signals are blocked.

4. For small-signal simulations such as AC and S-parameter analyses, AmplifierS2D uses the small-signal S-parameters of the S2D file only. For S-parameter analysis, the nominal frequency must be set as SSFreq=auto, where it implies freq, the ambient simulation frequency. The GCFreq parameter should be left empty for small-signal simulations. Linear interpolation of S-parameters is performed within the small-signal range to emulate behavior at a frequency not explicitly registered in the data file. Linear extrapolation is enabled but not recommended outside the small-signal frequency range because modeling accuracy cannot be guaranteed outside the data points of the S2D file.

5. For large-signal frequency analysis such as those based on Harmonic Balance, the small-signal S-parameters are used in conjunction with gain compression information in the GCOMPx block. Only odd-order harmonics are modeled.
during such large-signal simulations because information about even-order harmonics cannot be derived from the GCOMPx specification.

6. If gain compression information is presented in the GCOMP7 format at multiple frequencies, the AmplifierS2D GCFreq parameter must be assigned to one of the explicitly defined large-signal frequencies of the *.s2d file. The data model does not interpolate across large-signal frequencies so any violation of this requirement will result in termination of the simulation. This model resolves conflicts between user-assigned GCFreq values and large-signal frequency points in GCOMP7-based S2D files as shown in Table 8-2.

Table 8-2. Using GCFreq to Resolve GCOMP7 Frequency Conflicts

<table>
<thead>
<tr>
<th>Number of GCOMP7 blocks</th>
<th>Large signal frequencies (s2dfreq) specified in S2D file?</th>
<th>GCFreq specified on AmplifierS2D?</th>
<th>GCFreq = one s2dfreq?</th>
<th>Simulation Result. Compression data used at ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N/A</td>
<td>any</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>no</td>
<td>no</td>
<td>N/A</td>
<td>simulation frequency</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>N/A</td>
<td>GCFreq</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>no</td>
<td>N/A</td>
<td>s2dfreq</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>N/A. Simulation terminated with error message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>yes</td>
<td>GCFreq = s2dfreq</td>
</tr>
<tr>
<td>&gt; 1</td>
<td>no</td>
<td>no</td>
<td>N/A</td>
<td>simulation frequency</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>N/A</td>
<td>GCFreq. Use first available power sweep from GCOMP7</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>no</td>
<td>N/A</td>
<td>N/A. Simulation terminated with error message.</td>
</tr>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>N/A. Simulation terminated with error message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>yes</td>
<td>GCFreq = s2dfreq</td>
</tr>
</tbody>
</table>

7. For good performance using AmplifierS2D, include multiple GCOMP7 blocks at closely-spaced frequencies.

8. AmplifierS2D is equipped to index into the small-signal profile corresponding to the value of one user-defined variable using VarName and VarValue parameters. For example, if the *.s2d file contains small-signal scattering and noise parameters at various values of temperature where "temp" can take the discrete values 20, 30 or 40, then setting VarName="temp" and VarValue=30 will ensure that the data points corresponding to 30°C are used during simulation. It is important to note that the sequential variation of VAR temp through the S2D file must be monotonic and the statement VAR temp=x precede each relevant ACDATA block.
9. Although only one arbitrary VAR variable can be directly read and used by the AmplifierS2D component, the addition of a DataAccessComponent (DAC) can provide powerful support for accessing multi-dimensional S2D data as shown in the example design BEH_S2D_Access_multidimension.dsn in the example project AmplifierS2DandP2D_prj.

10. The S21 values expressed in the GCOMP7 block are not absolute values of S21 at the various input power, PIN levels. They are differential values in dB domain between power-swept S-parameters at the frequency of the GCOMP7 block and the small-signal S-parameters at the same frequency in the ACDATA block of the *.s2d file. The formats in which ACDATA and GCOMP7 data are represented in the same file may differ and may be in any of the forms DB / MA / RI. When calculating GCOMP7 values manually exercise care to convert absolute values of both large- and small-signal S-parameters to the dB domain before subtracting the latter from the former to obtain the numeric values in the GCOMP7 section.

11. AmplifierS2D is implemented using the SML Amplifier model. However, the gain compression information available via an S2D file is limited to the forward transmission characteristic defined by S21. If an application requires the knowledge and use of all four 2-port S-parameter variations but not the use of harmonics, consider using the AmplifierP2D model. If even-order harmonics must be modeled but sweeps of frequency are not required, consider using the AmpH1H2 model instead.
AmpLoadPull (SDD Load-Pull Amplifier)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset†</td>
<td>Name of dataset containing load-pull data for this amplifier model.</td>
<td></td>
<td>“dataset.ds”</td>
</tr>
</tbody>
</table>

† File-based data is also supported via use of the DataAccessComponent.

Notes/Equations

1. For amplifier designers, the AmpLoadPull and LoadPullSetup components address the issue of output match in one or more frequency bands. This is often investigated via load-pull contours indicating the load impedances, which, when presented to the output of an amplifier with a given source impedance and power, cause a certain power to be delivered to the load. The LoadPullSetup component extracts a dataset, given ranges and steps for input power, output reflection coefficient magnitude and output reflection coefficient angle. AmpLoadPull uses this dataset and allows fast behavioral amplifier simulations for all input power and output reflection coefficient values in the specified ranges.

2. This amplifier model uses the load-pull ADS dataset generated by the LoadPullSetup component. This behavioral profile can be referenced using the Dataset parameter of the AmpLoadPull component.

3. AmpLoadPull is useful in modeling nonlinear amplifiers with or without input/output matching networks. This model can be used in all types of simulations.

4. When using this model in other simulations, ensure the input power and the load value are within the range of the original data.

5. The use of this device and its extractor component is highlighted in the example project AmpLoadPull_prj.
System Data Models

Balun3Port (Balun, 3-port)

Symbol

Parameters
None

Notes/Equations

1. Balun3Port realizes the ideal transformation between a balanced differential-mode signal and unbalanced, single-ended signals. It can be useful to connect a source to a differentially fed circuit, although it does ignore common-mode effects.

2. Balun3Port realizes the voltage and current transformations given by:
   \[ v_d = v_+ - v_- \]
   \[ i_+ = -i_- = -i_d \]
   where
   \[ v_d/i_d = \text{the differential mode voltage/current at pin } d \]
   \[ v_+/i_+ = \text{the single line voltage/current at pin } + \]
   \[ v_-/i_- = \text{the single line voltage/current at pin } - \]
   The minus signs in the current definitions are due to the standard definition of currents directed into the Balun3Port component.

3. An equivalent functionality can be realized with a Balun4Port that has the common-mode pin grounded. However, the Balun3Port provides better convergence properties.

4. Balun3Port is bi-directional. When fed at the differential-mode pin, it realizes the transformations:
   \[ v_+ = -v_- = (v_d)/2 \]
   \[ i_+ = -i_- = -i_d \]
5. Examples of using Balun3Port to convert between (unbalanced) ADS sources
and balanced circuits are provided in the ADS examples directory; access these
examples from the ADS Main window > File > Example Project.

- **RFIC > MixerDiffMode_prj** demonstrates the use of Balun4Port to present
differential-mode sources (as well as common mode biases) to the RF and LO
inputs. It also shows the use of Balun3Port to single-ended
(differential-mode) IF output, which is needed to properly calculate the noise
figure.

- **BehavioralModels > DifferentialModels_prj** demonstrates the use of
Balun3Port and Balun4Port in conjunction with single-ended System-Data
Models in order to create a data-based behavioral model of a differentially fed
mixer.
System Data Models

Balun4Port (Balun, 4-port)

Symbol

Parameters

Notes/Equations

1. Balun4Port realizes the ideal transformation between balanced (differential- and common-mode) signals and unbalanced (single-ended) signals. It can be used to connect sources to a differentially fed circuit, particularly when modeling common-mode effects are important.

2. Balun4Port realizes voltage and current transformations given by:

\[ v_d = v_+ - v_- \]
\[ v_c = \frac{v_+ + v_-}{2} \]
\[ i_d = -\frac{i_+ - i_-}{2} \]
\[ i_c = -(i_+ + i_-) \]

where

- \( v_d/i_d \) = differential mode voltage/current at pin d
- \( v_c/i_c \) = common mode voltage/current at pin c
- \( v_+/i_+ \) = single line voltage/current at pin +
- \( v_-/i_- \) = single line voltage/current at pin -


The minus signs in the current definitions are due to the standard definition of currents directed into the Balun4Port component.

3. Balun4Port is bi-directional. It converts common/differential-mode signals into two single-ended signals, as well as converting two single-ended signals into common/differential mode signals.
4. If common-mode effects are not desired, Balun3Port provides an equivalent, but numerically more robust, result as grounding the common-mode pin of Balun4Port.

5. Examples of using Balun4Port to convert between (unbalanced) ADS sources and balanced circuits are provided in the ADS examples directory; access these examples from the ADS Main window > File > Example Project.

- RFIC > MixerDiffMode_prj demonstrates the use of Balun4Port to present differential-mode sources (as well as common mode biases) to the RF and LO inputs. It also shows the use of Balun3Port to single-ended (differential-mode) IF output, which is needed to properly calculate the noise figure.

- BehavioralModels > DifferentialModels_prj demonstrates the use of Balun3Port and Balun4Port in conjunction with single-ended System-Data Models in order to create a data-based behavioral model of a differentially fed mixer.
System Data Models

IQ_Demod_Data (IQ Demodulator Behavioral Model)

Symbol

![Symbol](image)

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Name of dataset generated by IQ_Demod_Setup.</td>
<td></td>
<td>“dataset.ds”</td>
</tr>
<tr>
<td>Freq</td>
<td>Modulation carrier frequency. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
</tbody>
</table>

Notes/Equations

1. IQ_Demod_Data is a data-based system model of a circuit-level I/Q demodulator. The circuit-level demodulator is characterized by a dataset generated by the system component IQ_Demod_Setup. The use of this component is highlighted in the example project IQ_Demod_prj.

2. The demodulation behaviors are extracted for a single carrier frequency. Only the demodulation distortions are characterized, so it is possible to set the frequency parameter in IQ_Demod_Data to differ from the value set in IQ_Demod_Setup. The demodulation will occur at the frequency set in IQ_Demod_Data. The extracted distortions to the baseband signal as it passes through IQ_Demod_Data will be applied at this frequency. IQ_Demod_Data does not model any dispersion in these distortions across the baseband bandwidth.

3. IQ_Demod_Data is designed for Circuit Envelope system verification where various filters typically eliminate unwanted frequency components. Therefore, IQ_Demod_Data makes no attempt to model harmonic components. The signals generated on the I and Q output ports are purely baseband signals. IQ_Demod_Data does not generate any frequency components at any carrier frequency.

4. In order for IQ_Demod_Data to produce an accurate model of the circuit-level demodulator characterized via IQ_Demod_Setup, the IQ_Demod_Setup Order parameter must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of about 5 should suffice.
while an Order in the range of 10-15 is recommended for highly non-linear circuits. Providing this criterion is met, the Order parameter for the simulation controller can be very low when using IQ_Demod_Data in a Circuit Envelope simulation. Typically, an Order of around 3 should suffice for accurate demodulation modeling. Note, however, that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.

5. For a circuit-level demodulator, the impedances presented by the input and output pins will generally be a complicated function of all state variables within the demodulator. For the IQ_Demod_Data, a certain simplification is necessary as no information about the environment in which the demodulator will later reside is known at the time of extraction. The input impedance at the RF input pin is extracted at the RF carrier frequency specified in IQ_Demod_Setup, and is a function of the input power. The impedances at the I and Q output pins are extracted at DC, and are modeled as constant impedances.

6. The extrapolation properties of IQ_Demod_Data above Pin-Stop are very poor. When using IQ_Demod_Data, ensure that the RF input signal does not exceed the Pin_Stop value set in IQ_Demod_Setup when the model was extracted. Extrapolation to signal levels below the Pin_Start value set in IQ_Demod_Setup will generally be good as long as the Pin_Start value lies within the linear operating range of the demodulator.

7. IQ_Demod_Data does not model noise.
IQ_Demod_Setup (IQ Demodulator Setup)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Modulation carrier frequency. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance analysis. Must be a positive integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Pin_Start†</td>
<td>Start value of input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>-40 dBm</td>
</tr>
<tr>
<td>Pin_Stop†</td>
<td>Stop value for input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>10 dBm</td>
</tr>
<tr>
<td>Pin_Step†</td>
<td>Interval for input power sweep during harmonic balance analysis.</td>
<td>dB</td>
<td>2 dB</td>
</tr>
</tbody>
</table>

† All power values are referenced to a 50 Ohm system. Pin_Stop >= the following values:

- \[ P_{\text{max,z}} + 10 \log_{10} (v_{\text{max}}) \] where \( v_{\text{max}} \) is the maximum peak voltage magnitude of the modulated signal.
- \[ 10 + 20 \cdot \log_{10} (P_{\text{max,z}}) \] where \( P_{\text{max,z}} \) is the maximum modulated signal power in dBm for a z Ohm system.

Notes/Equations

1. IQ_Demod_Setup performs a swept Harmonic Balance simulation of a circuit-level I/Q demodulator and generates a dataset for subsequent use by the data-based system model IQ_Demod_Data. The use of this component is highlighted in the example project IQ_Demod_prj.

2. In order for IQ_Demod_Data to produce an accurate model of the circuit-level demodulator characterized by IQ_Demod_Setup, the Order parameter for IQ_Demod_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of approximately 5 should suffice, while an Order in the range of 10-15 is recommended for highly non-linear circuits.

3. The demodulation behaviors are extracted for a single carrier frequency and over the range of modulated signal power specified by Pin_Start and Pin_Stop. These power values are specified in dBm with respect to a 50 Ohm system. For
an accurate IQ_Demod_Data behavioral model, Pin_Stop must be set at least as large as the expected maximum modulated signal power level. Refer to the Parameters table footnotes for conversion if working in other than a 50-Ohm system.

4. Push into IQ_Demod_Setup in a Schematic window for a view of the swept Harmonic Balance simulation controllers. This component can be copied and modified to suit individual needs; typically, this would entail changing the Harmonic Balance controller HB1 to achieve more efficient simulations. For more information, refer to the Harmonic Balance Simulation manual.

5. The extracted ADS dataset is assigned the name of the extraction design by default.
System Data Models

**IQ_Mod_Data (IQ Modulator Behavioral Model)**

**Symbol**

![Symbol Image]

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Name of dataset generated by IQ_Mod_Setup.</td>
<td></td>
<td>&quot;dataset.ds&quot;</td>
</tr>
<tr>
<td>Freq</td>
<td>Modulation carrier frequency.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
</tbody>
</table>

**Notes/Equations**

1. *IQ_Mod_Data* is a data-based system model of a circuit-level I/Q modulator. The circuit-level modulator is characterized by a dataset generated by the system component *IQ_Mod_Setup*. The use of this component is highlighted in the example project *IQ_Mod_prj*.

2. The modulation behaviors are extracted for a single carrier frequency. Only the modulation distortions are characterized, so it is possible to set the frequency parameter in *IQ_Mod_Data* to differ from the value set in *IQ_Mod_Setup*. The modulation will occur at the frequency set in *IQ_Mod_Data*. The extracted distortions to the baseband signal as it passes through *IQ_Mod_Data* will be applied at this frequency. *IQ_Mod_Data* does not model any dispersion in these distortions across the baseband bandwidth.

3. *IQ_Mod_Data* is designed for Circuit Envelope system verification where various filters typically eliminate unwanted frequency components. Therefore, *IQ_Demod_Data* makes no attempt to model harmonic components. The signal generated on RF output port is only a baseband signal modulated onto the carrier frequency. *IQ_Mod_Data* does not generate any frequency components at DC or at any harmonic of the carrier frequency.

4. In order for *IQ_Mod_Data* to produce an accurate model of the circuit-level demodulator characterized via *IQ_Mod_Setup*, the Order parameter for *IQ_Mod_Setup* must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of approximately 5 should suffice, while an Order in the range of 10-15 is
recommended for highly non-linear circuits. Providing this criterion is met, the Order parameter for the simulation controller can be very low when using IQ_Mod_Data in a Circuit Envelope simulation. Typically, an Order of approximately 3 should suffice for accurate demodulation modeling. Note, however, that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.

5. For a circuit-level modulator, the impedances presented by the input and output pins will generally be a complicated function of all state variables within the modulator. For the IQ_Mod_Data, a certain simplification is necessary as no information about the environment in which the modulator will later reside is known at the time of extraction. Specifically, IQ_Mod_Data presents a constant inward-looking impedance at the input and output pins. The input impedances at the baseband input pins are extracted at DC, while the impedance at the RF output pin is extracted at the RF carrier frequency specified in IQ_Mod_Setup.

6. The extrapolation properties of IQ_Mod_Data above Pin-Stop are very poor. When using IQ_Mod_Data, please ensure that the input baseband signal does not exceed the Pin_Stop value set in IQ_Mod_Setup when the model was extracted. Note that the Pin_Stop value is in reference to the complex baseband signal, which is typically a factor of sqrt(2), or 3 dB, larger than the power of the signal at either individual input pin. Extrapolation to signal levels below the Pin_Start value set in IQ_Mod_Setup will generally be good as long as the Pin_Start value lies within the linear operating range of the modulator.

7. IQ_Mod_Data does not model noise.
System Data Models

**IQ_Mod_Setup (IQ Modulator Setup)**

![Symbol](image)

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Modulation carrier frequency. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance analysis. Must be a positive integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Pin_Start†</td>
<td>Start value of input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>-40 dBm</td>
</tr>
<tr>
<td>Pin_Stop†</td>
<td>Stop value for input power sweep for harmonic balance analysis.</td>
<td>dBm</td>
<td>10 dBm</td>
</tr>
<tr>
<td>Pin_Step†</td>
<td>Interval for input power sweep during harmonic balance analysis.</td>
<td>dB</td>
<td>2 dB</td>
</tr>
</tbody>
</table>

† All power values are referenced to a 50-Ohm system. Pin_Stop >= the following values:

\[ P_{\text{max},z} + 10 \log_{10}(z/50) \]

where \( P_{\text{max},z} \) is the maximum modulated signal power in dBm for a \( z \) Ohm system.

### Notes/Equations

1. IQ_Mod_Setup performs a swept Harmonic Balance simulation of a circuit-level I/Q modulator and generates a dataset for subsequent use by the data-based system model IQ_Mod_Data. The use of this component is highlighted in the example project IQ_Mod_prj.

2. In order for IQ_Mod_Data to produce an accurate model of the circuit-level demodulator characterized by IQ_Mod_Setup, the Order parameter for IQ_Mod_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of approximately 5 should suffice, while an Order in the range of 10-15 is recommended for highly non-linear circuits.

3. The modulation behaviors are extracted for a single carrier frequency and over the range of modulation signal power specified by Pin_Start and Pin_Stop. These power values are specified in dBm with respect to a 50-Ohm system. For
an accurate IQ_Mod_Data behavioral model, Pin_Stop must be set at least as large as the expected maximum modulated signal power level. Refer to the Parameters table footnotes for conversion if working in other than a 50-Ohm system.

4. Push into IQ_Mod_Setup in a Schematic window for a view of the swept Harmonic Balance simulation controllers. This component can be copied and modified to suit individual needs; typically, this would entail changing the Harmonic Balance controller HB1 to achieve more efficient simulations. For more information, refer to the Harmonic Balance Simulation manual.

5. The extracted ADS dataset is assigned the name of the extraction design by default.
System Data Models

LoadPullSetup (Load Pull Setup)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Frequency of extraction. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance simulation inside extractor. Must be an integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Pin_Start</td>
<td>Start value for sweep of large-signal input power. Must be a real number.</td>
<td>dBm</td>
<td>-50 dBm</td>
</tr>
<tr>
<td>Pin_Stop</td>
<td>Stop value for sweep of large-signal input power. Must be a real number.</td>
<td>dBm</td>
<td>20 dBm</td>
</tr>
<tr>
<td>Pin_Step</td>
<td>Interval for linear sweep of large-signal input power. Must be a positive real number.</td>
<td>dB</td>
<td>10 dB</td>
</tr>
<tr>
<td>GamAng_Start</td>
<td>Start value for angle of output reflection coefficient. Must be a real number.</td>
<td>deg</td>
<td>-180 deg</td>
</tr>
<tr>
<td>GamAng_Stop</td>
<td>Stop value for angle of output reflection coefficient. Must be a real number.</td>
<td>deg</td>
<td>180 deg</td>
</tr>
<tr>
<td>GamAng_Step†</td>
<td>Interval for angle of output reflection coefficient. Must be a real number.</td>
<td>deg</td>
<td>20 deg</td>
</tr>
<tr>
<td>GamMag_Start</td>
<td>Start value for magnitude of output reflection coefficient. Must be a non-negative real number.</td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>GamMag_Stop</td>
<td>Stop value for magnitude of output reflection coefficient. Must be a non-negative real number.</td>
<td></td>
<td>0.9</td>
</tr>
<tr>
<td>GamMag_Step†</td>
<td>Interval for magnitude of output reflection coefficient. Must be a non-negative real number.</td>
<td></td>
<td>0.1</td>
</tr>
</tbody>
</table>

† Only linear sweeps using intervals of GamAng_Step, GamMag_Step are permitted using LoadAmpPull.

Notes/Equations

1. For amplifier designers, the AmpLoadPull and LoadPullSetup components address the issue of output match in one or more frequency bands. This is often investigated via load-pull contours indicating the load impedances that, when presented to the output of an amplifier with a given source impedance and power, cause a certain power to be delivered to the load. The LoadPullSetup
component extracts an ADS dataset, given ranges and steps for input power, output reflection coefficient magnitude and output reflection coefficient angle. AmpLoadPull uses this dataset and allows fast behavioral amplifier simulations for all input power and output reflection coefficient values in the specified ranges. The use of this component is highlighted in the example project AmpLoadPull_prj.

2. Although the behavioral data collected via LoadPullSetup is found using Harmonic Balance simulations, the subsequent behavioral amplifier simulations are not restricted to Harmonic Balance. In fact, the behavioral amplifier is assumed to be used for subsequent system verification (BER, ACPR etc. for Tx, Rx) within a Circuit Envelope framework. In such applications, various filters typically eliminate unwanted DC and harmonic components. Therefore, the load-pull suite was not designed to predict such components. This means that

• for a 1-tone Harmonic Balance simulation at frequency f, only the frequency component at f (fundamental) is retained. All others will be invalid or zero as they are assumed to be filtered later.

• for a 2-tone Harmonic Balance simulation at frequencies f1 and f2 (practically very close), only the frequency components at f1 and f2 (fundamentals) and those at 2 × f1-f2 and 2 × f2-f1 (intermodulation) are retained as these will be very close and cannot be assumed to be filtered. All others will be invalid or zero as they are assumed to be filtered later.

3. LoadPullSetup can be pushed into for a view of the implementation. If necessary, the component can be copied and modified to suit individual needs.

4. The extracted ADS dataset is assigned the name of the extraction design by default.
System Data Models

**MixerHBdata (2-Tone HB Mixer)**

**Symbol**

![Diagram of 2-Tone HB Mixer]

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Name of dataset for 2-tone harmonic balance mixer generated by MixerHBsetup.</td>
<td>“dataset.ds”</td>
<td></td>
</tr>
<tr>
<td>RFIndexExpr</td>
<td>RF tone mixing index defined in the MixerHBsetup component at the time of dataset extraction. Typically set to “Mix(2)”.</td>
<td></td>
<td>“Mix(2)”</td>
</tr>
<tr>
<td>LOIndexExpr</td>
<td>LO tone mixing index defined in the MixerHBsetup component at the time of dataset extraction. Typically set to “Mix(1)”.</td>
<td></td>
<td>“Mix(1)”</td>
</tr>
<tr>
<td>IFwaveExpr</td>
<td>IF wave variable defined in the MixerHBsetup component at the time of dataset extraction. Typically set to “b2”.</td>
<td></td>
<td>“b2”</td>
</tr>
<tr>
<td>IMdata</td>
<td>Intermodulation data flag. Can be -1 or 1 only.</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>ConvGain</td>
<td>Conversion gain multiplier from RF port (Port 1) to IF port (Port 2).</td>
<td>dbpolar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP11</td>
<td>RF port reflection coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP12</td>
<td>IF port to RF port leakage coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP13</td>
<td>LO port to RF port leakage coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP21</td>
<td>RF port to IF port transmission/leakage coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP22</td>
<td>IF port reflection coefficient.</td>
<td>polar(0,180)</td>
<td></td>
</tr>
<tr>
<td>SP23</td>
<td>LO port to IF port transmission/leakage coefficient</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP31</td>
<td>RF port to LO port leakage coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP32</td>
<td>IF port to LO port leakage coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP33</td>
<td>LO port reflection coefficient.</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>NF</td>
<td>Input double side band noise figure.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>NFmin</td>
<td>Minimum double sideband noise figure at Sopt.</td>
<td>dB</td>
<td></td>
</tr>
</tbody>
</table>

† Typically not edited by user. Should be synchronized with expressions registered in dataset created using MixerHBsetup.
†† Set to -1 for lower sideband and 1 for upper sideband. Refer to note 4 for details.
††† These parameters can be reported in any of the following complex number formats: x + jy, polar(x,y), dbpolar(x,y), vswrpolar(x,y)
‡ Transmission/leakage implies that the value of this parameter is used as an additive supplement to gain expressions between RF->IF and LO->IF ports.
‡‡ Refer to note 6 for details.
MixerHBdata (2-Tone HB Mixer) 8-33

Notes/Equations

1. MixerHBdata is a data-based system model of a circuit-level mixer. The dataset used by MixerHBdata is generated from the circuit-level mixer by the behavioral extractor component MixerHBsetup (available in the System-Data Models palette). Ports 1, 2 and 3 on MixerHBdata correspond to the RF, IF and LO ports of MixerHBsetup.

2. MixerHBdata can be used in Harmonic Balance, Circuit Envelope, and Transient simulations.

3. MixerHBdata is currently configured to produce IF output with zero phase regardless of the IF output phase represented in the MixerHBsetup extracted dataset.

4. A dataset extracted using the MixerHBsetup component contains both upper and lower sideband harmonics and intermodulation products. By formal definition the upper sideband contains frequencies above the higher of the two mixing frequencies. For example, in a typical RFIC receiver, the center frequency is the RFfreq because LOfreq < RFfreq. In the MixerHBdata component the IMdata parameter is used to prompt the use of only one of these

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_{opt}†††</td>
<td>Optimum source reflection for minimum noise figure.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R_{n}</td>
<td>Equivalent noise resistance. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>Z_{1}</td>
<td>Reference impedance for RF port.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>Z_{2}</td>
<td>Reference impedance for IF port.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>Z_{3}</td>
<td>Reference impedance for LO port.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>P_{RFnom} ‡</td>
<td>Power incident at the RF port. Must be manually synchronized with source signal value at this port.</td>
<td>dBm</td>
<td>-50 dBm</td>
</tr>
<tr>
<td>P_{LOnom} ‡</td>
<td>Power incident at the LO port. Must be manually synchronized with source signal value at this port.</td>
<td>dBm</td>
<td>-5 dBm</td>
</tr>
<tr>
<td>M_{RF} ‡</td>
<td>Number of RF harmonics to be considered from dataset.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>N_{LO} ‡</td>
<td>Number of LO harmonics to be considered from dataset.</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

† Typically not edited by user. Should be synchronized with expressions registered in dataset created using MixerHBsetup.
‡ ‡ ‡ Set to -1 for lower sideband and 1 for upper sideband. Refer to note 4 for details.
†† †† †† These parameters can be reported in any of the following complex number formats: x + j y, polar(x,y), dbpolar(x,y), vswrpolar(x,y)
‡ Transmission/leakage implies that the value of this parameter is used as an additive supplement to gain expressions between RF->IF and LO->IF ports.
‡‡ ‡‡ Refer to note 6 for details.
two bands for deriving an internal polynomial model for frequency response. Once this model has been constructed, it is used to represent behavior of both sidebands in the MixerHBdata simulation output.

5. If a 2-tone Harmonic Balance simulation is to be performed on a MixerHBdata component using $F_{\text{req1}}=\text{LOfreq}$ and $F_{\text{req2}}=\text{RFfreq}$ in the simulator component, then setting $\text{IMdata}=1$ selects the lower sideband frequencies and $\text{IMdata}=1$ selects the upper sideband frequencies from the dataset. The harmonics and intermodulation products are defined as $|m \times RFfreq - n \times LOfreq|$ in the lower sideband and $|m \times RFfreq + n \times LOfreq|$ in the upper sideband, where $m$ and $n$ are integers whose absolute values are bounded by $M_{RF}$ and $N_{LO}$ (the maximum harmonic order parameters of the MixerHBdata component).

6. By the convention that $-M_{RF} \leq m \leq M_{RF}$ and $-N_{LO} \leq n \leq N_{LO}$, and the assumption that $M_{RF}$ and $N_{LO}$ are of comparable magnitudes, the difference frequencies $|m \times RFfreq - n \times LOfreq|$ are guaranteed to be lower than RFfreq and the summed frequencies $|m \times RFfreq + n \times LOfreq|$ are guaranteed to be higher than RFfreq. Thus, the IMdata parameter helps in sideband selection and should be assigned values of -1 or 1 only. For proper modeling $M_{RF}$ and $N_{LO}$ of the MixerHBdata component should be set to the value of RForder and LOorder settings of the MixerHBsetup that was used to extract the behavioral dataset.

7. The use of this component is highlighted in the example project MixerHBdata_prj.
MixerHBsetup (2-Tone HB Mixer Setup)

Symbol

![Symbol Image]

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFfreq</td>
<td>RF frequency. Denotes incoming message signal whether it is intended for up or down conversion. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>LOfreq</td>
<td>LO frequency. Denotes frequency of local oscillator or carrier frequency. Must be a non-negative real number.</td>
<td>Hz</td>
<td>1.1 GHz</td>
</tr>
<tr>
<td>RFpwr</td>
<td>Power incident at the RF port. Must be manually synchronized with source signal value at this port.</td>
<td>dBm</td>
<td>-50 dBm</td>
</tr>
<tr>
<td>LOpwr</td>
<td>Power incident at the LO port. Must be manually synchronized with source signal value at this port.</td>
<td>dBm</td>
<td>-50 dBm</td>
</tr>
<tr>
<td>RFord</td>
<td>Number of RF harmonics allowed to participating in mixing. Must be a positive integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>LOord</td>
<td>Number of LO harmonics allowed to participating in mixing. Must be a positive integer greater than 0.</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

Notes/Equations

1. MixerHBsetup extracts the behavior of a circuit-level mixer using a Harmonic Balance simulation. The extracted dataset can be subsequently used by the corresponding data-based system model MixerHBdata. Ports RF, IF and LO ports of the MixerHBsetup component correspond to the ports 1, 2 and 3 on the MixerHBdata component.

2. MixerHBsetup can be pushed into for a view of the extractor implementation including access to the built-in Harmonic Balance simulator component. Note that equations and variables of the extractor subcircuit are referenced by name from parameters of the MixerHBdata component. Any user-initiated changes made to a specific instance of MixerHBsetup must be reflected in a complementary instance of MixerHBdata for correct dataset indexing for behavioral modeling.
System Data Models

3. The use of this component is highlighted in the example project MixerHBdata_prj.

4. The extracted ADS dataset is assigned the name of the extraction design by default.
MixerIMT (Obsolete Intermodulation Table Mixer)

Symbols:

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS_SideBand</td>
<td>Produce UPPER or LOWER sideband at output port for linear analysis.</td>
<td></td>
<td>UPPER</td>
</tr>
<tr>
<td>ConvGain</td>
<td>Conversion gain multiplier from RF port (Port 1) to IF port (Port 2).</td>
<td>dBpolar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP11</td>
<td>RF port reflection coefficient.</td>
<td>polar(0,0)</td>
<td></td>
</tr>
<tr>
<td>SP22</td>
<td>IF port reflection coefficient.</td>
<td>polar(0,180)</td>
<td></td>
</tr>
<tr>
<td>SP33</td>
<td>LO port reflection coefficient.</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>NF</td>
<td>Input double sideband noise figure. Must be a non-negative real number.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>NFmin</td>
<td>Minimum double sideband noise figure at Sopt. Must be a positive real number.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Sopt†</td>
<td>Optimum source reflection for minimum noise figure. Must have magnitude less than 1.0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rn</td>
<td>Equivalent noise resistance. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>Reference impedance for RF port. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Reference impedance for IF port. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>Reference impedance for LO port. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>InThresh</td>
<td>Voltage threshold for signal input at RF port. Must be a positive real number.</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>LoThresh</td>
<td>Voltage threshold for signal input at LO port. Must be a positive real number.</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>IMT_File</td>
<td>File containing intermodulation table. Accepted formats MDIF: IMT, P2D, S2D.</td>
<td></td>
<td>&quot;imtfile.imt&quot;</td>
</tr>
</tbody>
</table>

† These parameters can be reported in any of the following complex number formats: \( x + jy \), polar\((x,y)\), dbpolar\((x,y)\), vswrpolar\((x,y)\)

Notes/Equations:

1. This component is included for the convenience of customers who have existing designs with the MixerIMT model. New instances can be accessed by typing
MixerIMT into the Component History field in the Schematic window, pressing Enter, and moving your cursor to the drawing area to place it.

2. MixerIMT does not function in Transient simulations; MixerIMT blocks baseband signals in Circuit Envelope simulations; MixerIMT does not produce intermodulation among RF input tones (refer to note 9). Replace MixerIMT with MixerIMT2 for these functions.

3. The default value of S12=0 to avoid creating cyclic interdependencies between the RF and IF ports.

4. If no IMT file is referenced, the output (IF) signal is the product of the input RF and LO spectral tones only.

5. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

\[
\frac{R_n}{Z_0} \geq \frac{T_0}{4} \left( \frac{1 + S_{opt}}{1 - S_{11}} \right)^2 \frac{T_4}{|1 - S_{opt} S_{11}|^2}
\]

Rn will be reset to a value that meets this criteria if it does not satisfy this condition.

6. Use the function polar(mag,ang), or dbpolar(dB,ang), or VSWRpolar(VSWR, ang) to convert these specifications into a complex number.

7. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple (mag, intercept, period, variable); for example ripple(0-1, 0-1, 10 MHz, Freq).

Example: $S_{21}=\text{dbpolar}(10+\text{ripple}(),0.)$

8. This model blocks dc.

9. MixerIMT produces intermodulation products at frequencies

\[|m \times \text{LO} \pm n \times \text{Signal}|
\]

where

\[-M \leq m \leq M \text{ and } -N \leq n \leq N\]

Also, M and N are the orders for LO and RF signal frequencies set in the Harmonic Balance or Circuit Envelope simulation controllers.

If there are multiple frequencies at the RF and LO ports, there will be a superposition of the RF-LO intermodulation tones. For example, given the
frequencies LO1 and LO2 at the LO port, and Signal1 and Signal2 at the RF port, there will be intermodulation products at the frequencies

\[ |m \times \text{LO1} \pm n \times \text{Signal1}|, |m \times \text{LO1} \pm n \times \text{Signal2}|, \]

\[ |m \times \text{LO2} \pm n \times \text{Signal1}|, |m \times \text{LO2} \pm n \times \text{Signal2}| \]

where

\[-M_{\text{RF}} \leq m \leq M_{\text{RF}} \text{ and } -N_{\text{LO}} \leq n \leq N_{\text{LO}}\]

There will be no products at \(|\text{LO1} \pm \text{LO2}|, |\text{Signal1} \pm \text{Signal2}|\) or any other RF or LO self-modulation products.

**Intermodulation Product Table File Used with MixerIMT**

The intermodulation product table (.imt) file is a user-defined table of mixer intermodulation (IM) products between the LO and input signal. It relates your mixer IM output level to the output signal level. The output signal is a direct mapping of each input tone with each LO tone. Interaction between input signal tones are not characterized in this model. Example IMT files are provided in Figure 8-1 and Figure 8-2. After selecting the IMT_FILE parameter, click on the Copy template button in the dialog box to select a file.

In the IMT tables:

- The vertical row number N (0, 1, to 15) indicates the harmonic of the signal used in deriving the spurious output signal.

- The horizontal column number M (0, 1, to 15) indicates the harmonic of the local oscillator used in deriving the spurious output signal.

- In row 2, column 4, the data is 13. This means that for an input signal at −10 dBm input, with an LO drive of +7 dBm, an output spurious signal will occur at \(3 \times \text{LO} \pm 1 \times \text{signal}\), with a level that is 13 dB below the fundamental output signal.

- If the input signal differs from the −10 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding \((N-1) \times X\) dB to it. This manner of adjustment is good for input power levels up to 5 dB greater than the reference signal power.

- If the local oscillator signal differs from the +7 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding it by \(M \times X\) dB to it. This manner of adjustment is good for local
oscillator power levels from the reference level minus 10 dB to the reference level plus 3 dB.

- If items are missing from the IMT table, a triangular table, they are loaded as 200 dB down from input reference.

```
BEGIN IMT_DATA
! DLB1.IMT
! Intermodulation table for double balanced mixer #1
! Signal Level (dBm)  Lo Level (dBm)
# IMT  (-10  7)!
! M x Lo (Horizontal)  N x Signal (Vertical)
%  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 !
!  99 26 35 39 50 41 53 49 51 45 65 55 75 65 85 99 !row0
24 0 35 13 40 24 45 28 49 35 55 45 65 55 99 !row1
73 73 74 70 71 64 69 64 69 65 75 75 85 99 !row2
67 64 69 50 77 47 74 44 74 45 75 55 99 !row3
86 90 86 88 85 86 90 85 85 99 !row4
90 80 90 71 90 68 90 65 88 65 99 !row5
90 90 90 90 90 90 90 90 90 99 !row6
90 90 90 90 87 90 90 90 90 99 !row7
99 95 99 99 95 99 99 99 99 99 !row8
90 95 99 99 99 99 99 99 99 99 !row9
END
```

Figure 8-1. dlbl.imt File Example
BEGIN IMT_DATA
! DLB2.IMT
! Intermodulation table for double balanced mixer #2
! Signal Level (dBm)  Lo Level (dBm)
# IMT  (-10  17)
!  M x Lo (Horizontal)  N x Signal (Vertical)
% 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
! 99 39 42 46 58 37 65 49 75 62 72 61 70 57 87 60 !row0
 25  0 39 11 50 16 59 19 59 43 63 52 70 57 73 99 !row1
 68 67  7 67 80 66 82 66 83 72 84 72 82 70 99 !row2
 63 58 65 60 65 55 64 54 66 57 85 54 70 99 !row3
 96 80 96 80 95 82 98 78 90 95 95 95 95 99 !row4
 93 73 87 72 88 66 85 64 82 75 95 99 !row5
 99 95 99 95 95 95 95 95 95 95 95 95 95 !row8
 90 95 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 !row9

Figure 8-2. dbl2.imt File Example
System Data Models

MixerIMT2 (Intermodulation Table Mixer)

Symbols:

![Diagram of MixerIMT2]

### Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvGain</td>
<td>Conversion gain multiplier from RF port (Port 1) to IF port (Port 2).</td>
<td></td>
<td>dbpolar(0,0)</td>
</tr>
<tr>
<td>SP11</td>
<td>RF port reflection coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP12</td>
<td>IF port to RF port leakage coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP13</td>
<td>LO port to RF port leakage coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP21</td>
<td>RF port to IF port transmission/leakage†† coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP22</td>
<td>IF port reflection coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP23</td>
<td>LO port to IF port transmission/leakage†† coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP31</td>
<td>RF port to LO port leakage coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP32</td>
<td>IF port to LO port leakage coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>SP33</td>
<td>LO port reflection coefficient.</td>
<td></td>
<td>polar(0,0)</td>
</tr>
<tr>
<td>NF</td>
<td>Input double side band noise figure. Must be a non-negative real number.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>NFmin</td>
<td>Minimum double sideband noise figure at Sopt. Must be a positive real number.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Sopt</td>
<td>Optimum source reflection for minimum noise figure. Must have magnitude less than 1.0.</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Rn</td>
<td>Equivalent noise resistance. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>Reference impedance for RF port. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>Reference impedance for IF port. Must be a non-negative real number.</td>
<td>Ohm</td>
<td></td>
</tr>
</tbody>
</table>

† These parameters can be reported in any of the following complex number formats: \( x + jy \), polar\((x,y)\), dbpolar\((x,y)\), vswrpolar\((x,y)\)

†† Transmission/leakage implies that the value of this parameter is used as an additive supplement to gain expressions between RF->IF and LO->IF ports.

††† Cannot exceed respective orders of input signals within IMT table.
1. If NFmin, Sopt, and Rn are used to characterize noise, the following relation must be satisfied for a realistic model.

\[
\frac{R_n}{Z_0} \geq T \left( \frac{F_{\min} - 1}{1 + S_{\text{opt}}} \right)^2 \frac{(1 - |S_{11}|^2)}{|1 - S_{\text{opt}} S_{11}|^2}
\]

If the noise parameters attempt to describe a system that requires negative noise (due to Rn being too small), the negative part of the noise will be set to 0 and a warning message will be issued.

2. Use the function polar(mag,ang), or dbpolar(dB,ang), or vswrpolar(VSWR, ang) to convert these specifications into a complex number.

3. For an S-parameter or a noise figure sinusoidal ripple, use the function ripple (mag, intercept, period, variable); for example, ripple(0-1, 0-1, 10 MHz, Freq).

Example: S21=dbpolar(10+ripple(),0.)

4. MixerIMT2 produces intermodulation products at frequencies

\[
| m \times \text{LO} \pm n \times \text{Signal} |
\]

where

\[-M_{\text{RF}} \leq m \leq M_{\text{RF}} \text{ and } -N_{\text{LO}} \leq n \leq N_{\text{LO}}\]
If there are multiple frequencies at the RF and LO ports, there will be a superposition of the RF-LO intermodulation tones. For example, given the frequencies LO1 and LO2 at the LO port, and Signal1 and Signal2 at the RF port, there will be intermodulation products at the frequencies
\[ |m \times LO_1 \pm n \times Signal_1|, |m \times LO_1 \pm n \times Signal_2|, \]
\[ |m \times LO_2 \pm n \times Signal_1|, |m \times LO_2 \pm n \times Signal_2| \]
where
\[-M_{RF} \leq m \leq M_{RF} \text{ and } -N_{LO} \leq n \leq N_{LO}\]
In addition, there will be no products at \(|LO_1 \pm LO_2|, |Signal_1 \pm Signal_2|\) and other RF or LO self-modulation products.

5. MixerIMT2 can be used in all simulations except for Frequency Converting AC analysis.

6. MixerIMT2 leakage terms are specified in two separate ways: the IMT file entries, and the SPij parameters. For example, LO to IF leakage is given by the \((M=1, N=0)\) term in the IMT file as well as the SP23 parameter. When both specifications are given, the total leakage is given by the complex sum of the two specifications. It is generally advisable to leave \(SP23=SP13=0\); in this case, the LO to IF and the RF to IF leakages are specified by the IMT file.

**Intermodulation Product Table File Used with MixerIMT and MixerIMT2**

The intermodulation product table (.imt) file is a user-defined table of mixer intermodulation (IM) products between the LO and input signal. It relates your mixer IM output level to the output signal level. The output signal is a direct mapping of each input tone with each LO tone. Interaction between input signal tones are not characterized in this model. IMT file examples are shown in Figure 8-3 and Figure 8-4. After selecting the IMT File parameter, click on the Copy template button in the dialog box to select a file.

In the IMT tables:

- The vertical row number \(N\) (0, 1, to 15) indicates the harmonic of the signal used in deriving the spurious output signal.
- The horizontal column number \(M\) (0, 1, to 15) indicates the harmonic of the local oscillator used in deriving the spurious output signal.
• In row 2, column 4, the data is 13. This means that for an input signal at −10 dBm input, with an LO drive of +7 dBm, an output spurious signal will occur at $3 \times LO + 4\times - 1\times signal$, with a level that is 13 dB below the fundamental output signal.

• If the input signal differs from the −10 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding $(N-1) \times X$ dB to it. This manner of adjustment is good for input power levels up to 5 dB greater than the reference signal power.

• If the local oscillator signal differs from the +7 dBm reference power level listed at the top of the table by X dB, then the number in the table is adjusted by adding it by $M \times X$ dB to it. This manner of adjustment is good for local oscillator power levels from the reference level minus 10 dB to the reference level plus 3 dB.

• If items are missing from the IMT table, a triangular table, they are loaded as 200 dB down from the input reference.

BEGIN IMT_DATA
! DLE1.IMT
! Intermodulation table for double balanced mixer #1
! Signal Level (dBm)  Lo Level (dBm)
# IMT (−10  7)
!  M x Lo (Horizontal)  N x Signal (Vertical)
% 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
! 99 26 35 39 50 41 53 49 51 45 65 55 75 65 85 99 !row0
24 0 35 13 40 24 45 28 49 35 55 45 65 55 99 !row1
74 73 74 70 71 64 69 64 69 65 75 75 85 99 !row2
67 64 69 50 77 47 74 44 74 45 75 55 99 !row3
86 90 86 88 85 86 85 90 85 99 !row4
90 80 90 71 90 68 90 65 88 65 99 !row5
90 90 90 90 90 90 90 99 !row6
90 90 90 90 90 90 90 99 !row7
99 95 99 95 99 95 99 95 99 99 !row8
90 95 90 95 90 99 99 !row9
99 99 99 99 99 99 !row10
90 99 99 99 99 !row11
99 99 99 99 !row12
99 99 99 !row13
99 99 !row14
99 !row15
END

Figure 8-3. dbl1.imt File Example
BEGIN IMT_DATA
! DLB2.IMT
! Intermodulation table for double balanced mixer #2
! Signal Level (dBm)  Lo Level (dBm)
% IMT (-10 17)
! M x Lo (Horizontal)  N x Signal (Vertical)
! 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
! 99 39 42 46 58 37 65 49 75 62 72 61 70 57 87 60 !row0
25 0 39 11 50 16 59 19 59 43 63 52 70 57 73 99 !row1
68 67 76 67 80 66 82 66 83 72 84 72 82 70 99 !row2
63 58 65 60 65 55 64 54 66 57 85 54 70 99 !row3
96 80 96 80 98 92 88 78 90 95 95 95 99 !row4
93 73 87 72 88 66 85 64 82 75 95 99 !row5
99 79 99 78 99 78 99 81 99 99 !row7
99 95 99 95 99 99 95 95 95 95 95 !row8
90 95 90 90 90 99 90 90 99 !row9
90 99 90 95 90 !row11
99 99 99 99 99 !row12
90 99 90 99 !row13
99 99 !row14
99 99 !row15
END

Figure 8-4. db2.imt File Example

Example

The following example table is for small-signal input conditions for which the IM level is fixed by the type of mixer and LO drive. It is an example of a characteristic mixer at a specific LO drive level; it does not give IM results for multi-tone signal input. In the example, the IM product for 5×LO + 1×Signal is 24 dB below the fundamental output; the IM product for 2×LO + 3×Signal is 69 dB below the fundamental output.

MixerIMT2 also supports an IMT table that contains absolute dBm values at a particular RF level and a particular LO level. The IMT table is not necessarily 15x15. Following is such an example. Please ensure that IMTValueType=IMTValue in this case.

This example table shows that for an RF input of -30 dBm and a LO input of 0 dBm, the 1×1 product at output is -29 dBm. The LO feedthrough (0×1 product) is -7.8 dBm. The RF feedthrough (1×0 product) is -30.8 dBm etc.
<table>
<thead>
<tr>
<th>Signal Level (dBm)</th>
<th>LO level (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-30</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M x LO (Horizontal)</th>
<th>N x Signal (Vertical)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13</td>
</tr>
<tr>
<td>-110</td>
<td>-7.8 -35.7 -30.2 -53.2 -38.8 -39.5</td>
</tr>
<tr>
<td>-105.2</td>
<td>-79.2 -74.5 -92.3 -93 -99.2 -106.8</td>
</tr>
<tr>
<td>-110</td>
<td>-110 -110 -105.2 -110 -110 -110</td>
</tr>
<tr>
<td>-110</td>
<td>-110 -110 -110 -110 -110 -110</td>
</tr>
<tr>
<td>-110</td>
<td>-110 -110 -110 -110 -110 -110</td>
</tr>
</tbody>
</table>
VCA_Data (Voltage Controlled Amplifier)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Name of dataset generated by VCA_Setup model.</td>
<td></td>
<td>&quot;dataset.ds&quot;</td>
</tr>
</tbody>
</table>

Notes/Equations

1. VCA_Data is a data-based system model of a circuit-level VCA. The circuit level VCA is characterized by a dataset generated by the system component VCA_Setup. The use of this component is highlighted in the example project VCA_prj.

2. VCA_Data is the most advanced data-based system amplifier. If no special circumstances make AmplifierS2D, AmplifierP2D, AmpH1H2 or AmpLoadPull superior to VCA_Data, the latter model should be used for data-based system modeling of amplifiers. Since the VCA_Data does not have provisions for explicit control of frequency, order of harmonics or bias voltage at the instance level, user discretion is advised in ensuring that behavioral simulation and extraction simulation environments are closely matched for reliable modeling.

3. VCA_Setup and VCA_Data work equally well for amplifiers with or without a control voltage. For amplifiers without a control voltage, the control pin should be grounded.

4. VCA_Data is designed for Harmonic Balance or Circuit Envelope system verification (ACPR, BER, etc.) where various filters typically eliminate unwanted frequency components. Therefore, VCA_Data makes no attempt to model harmonic components. This means that for a 1-tone Harmonic Balance simulation at frequency f, only the frequency component at f (fundamental) is retained. All others will be invalid or zero as they are assumed to be filtered later.

For a 2-tone Harmonic Balance simulation at frequencies f1 and f2 (practically very close), only the frequency components at f1 and f2 (fundamentals) and those at 2 x f1-f2 and 2 x f2-f1 (intermodulation) are retained as these will be
very close and cannot be assumed to be filtered. All others will be invalid or zero as they are assumed to be filtered later. Similarly, for a Circuit Envelope simulation, narrowband modulated components around the carrier(s) will be retained while all others will be invalid or zero as they are assumed to be filtered later.

VCA_Data will also run in Transient but this is not a recommended simulation controller. Since VCA_Setup is based on Harmonic Balance analysis, the dataset created by VCA_Setup contains steady-state information only. Transient analysis can predict the correct steady-state response only and will almost certainly result in an incorrect transient response.

5. In order for VCA_Data to produce an accurate model of the circuit-level VCA characterized via VCA_Setup, the Order parameter for VCA_Setup must be large enough to prevent aliasing of higher-order frequency components. For mildly non-linear circuits, an Order of approximately 5 should suffice while a range of 10-15 is recommended for highly non-linear circuits. Provided this criterion is met, the Order parameter for the simulation controller can be very low when using VCA_Data in a Harmonic Balance or Circuit Envelope simulation. Typically, an Order of approximately 3 should suffice for accurate VCA modeling. However, note that accurate modeling of other components of the complete system may necessitate a larger value for this parameter.

6. For a circuit-level VCA, the impedances presented by the input, output and control pins will generally be highly complicated functions of all state variables within the circuit of which the VCA is part. For a data-based VCA, a simplification is necessary as no information about the environment in which the VCA will later reside is known at the time of extraction. Specifically, VCA_Data presents an input power and control voltage dependent impedance at the input, output and control pins, with the input and control impedances assuming an open-circuit output pin. The input and output impedances are voltage-to-current ratios at RF while the control impedance is a voltage-to-current ratio at DC.

7. Assuming proper sampling, the interpolation properties of VCA_Data are generally good but occasionally break down near the limits of the ranges to which input power and control voltage were constrained when generating the dataset using VCA_Setup. To safe-guard against such a breakdown, the upper and lower limits of these ranges can be extended about 6 dB in input power and 1 V in control voltage.
System Data Models

8. The extrapolation properties of VCA_Data are very poor. When using VCA_Data, please ensure that input power and control voltage are within the ranges to which input power and control voltage were constrained when generating the dataset using VCA_Setup.

9. VCA_Data does not model noise.

10. VCA_Setup and VCA_Data can be used for simulating an amplifier with an arbitrary swept control parameter, for example a temperature. Simply convert a 2-pin amplifier dependent on a given parameter, in this case Par, to a 3-pin amplifier with the third pin connected to a grounded DC voltage source whose strength equals Par. By sweeping the control voltage of VCA_Setup in the desired parameter range for Par and subsequently connecting the control pin of VCA_Data to a grounded DC voltage source, Par values in the appropriate range can be selected by simply setting the strength of this voltage source properly.
VCA_Setup (Voltage Controlled Amplifier Setup)

Symbol

Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Unit</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>Fundamental frequency.</td>
<td>Hz</td>
<td>1.0 GHz</td>
</tr>
<tr>
<td>Order</td>
<td>Order for Harmonic Balance simulation inside extractor.</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Pin_Start</td>
<td>Start value for sweep of large-signal input power.</td>
<td>dBm</td>
<td>-10 dBm</td>
</tr>
<tr>
<td>Pin_Stop</td>
<td>Stop value for sweep of large-signal input power.</td>
<td>dBm</td>
<td>15 dBm</td>
</tr>
<tr>
<td>Pin_Step</td>
<td>Interval for linear sweep of large-signal input power.</td>
<td>dB</td>
<td>5 dB</td>
</tr>
<tr>
<td>Vcontrol_Start</td>
<td>Start value for sweep of control voltage.</td>
<td>V</td>
<td>0.0 V</td>
</tr>
<tr>
<td>Vcontrol_Stop</td>
<td>Stop value for sweep of control voltage.</td>
<td>V</td>
<td>1.0 V</td>
</tr>
<tr>
<td>Vcontrol_Step</td>
<td>Interval for linear sweep of control voltage.</td>
<td>V</td>
<td>0.1 V</td>
</tr>
</tbody>
</table>

Notes/Equations

1. VCA_Setup performs a swept control voltage and swept input power simulation of a circuit-level VCA and generates a dataset for subsequent use by the data-based system model VCA_Data. The use of this component is highlighted in the example project VCA_prj.

2. VCA_Setup and VCA_Data work equally well for amplifiers with or without a control voltage. For amplifiers without a control voltage, the control pin should be grounded.

3. In order for VCA_Data to produce an accurate model of the circuit-level VCA characterized via VCA_Setup, the Order parameter for VCA_Setup must be large enough to prevent aliasing of higher-order frequency components. For
mildly non-linear circuits, an Order of approximately 5 should suffice while an Order in the range 10-15 is recommended for highly non-linear circuits.

4. Push into VCA_Setup in a Schematic window for a view of the swept Harmonic Balance simulation controllers. This component can be copied and modified to suit individual needs; typically, this would entail changing the Harmonic Balance controller HB1 to achieve more efficient simulations. For more information, refer to the Harmonic Balance Simulation manual.

5. When simulating an existing ADS2002 design containing a VCA_Setup instance in the current version of ADS, it is necessary to upgrade by substitution to a newer version to avoid generation of warning messages and corrupting the extracted dataset.

6. The extracted ADS dataset is assigned the name of the extraction design by default.
Index

A
AM_DemodTuned, 3-3
AM_ModTuned, 3-4
AmpH1H2, 8-4
AmpH1H2_Setup, 8-6
Amplifier, 1-3
Amplifier2, 1-11
AmplifierP2D, 8-8
AmplifierP2D_Setup, 8-11
AmplifierS2D, 8-13
AmplifierVC, 1-35
AmpLoadPull, 8-17
AmpSingleCarrier, 1-36
AntLoad, 4-2
Attenuator, 4-4

B
Balun3Port, 4-9, 8-18
Balun4Port, 4-11, 8-20
Balun6Port, 4-13
BPF_Bessel, 2-6
BPF_Butterworth, 2-7
BPF_Chebyshev, 2-8
BPF_Elliptic, 2-10
BPF_Gaussian, 2-11
BPF_PoleZero, 2-12
BPF_Polynomial, 2-14
BPF_RaisedCos, 2-15
BSF_Bessel, 2-18
BSF_Butterworth, 2-19
BSF_Chebyshev, 2-20
BSF_Elliptic, 2-21
BSF_Gaussian, 2-22
BSF_PoleZero, 2-23
BSF_Polynomial, 2-24
BSF_RaisedCos, 2-25

C
Circulator, 4-14
ClockLFSR, 6-3
Comparator, 6-2
CouplerDual, 4-16
CouplerSingle, 4-18

D
Differentiator, 6-7
DivideByN, 5-2
DPDT_Static, 6-8

F
FM_DemodTuned, 3-5
FM_ModTuned, 3-6
FreqMult, 1-37

G
Gyrator, 4-20

H
HPF_Bessel, 2-28
HPF_Butterworth, 2-29
HPF_Chebyshev, 2-30
HPF_Elliptic, 2-31
HPF_Gaussian, 2-32
HPF_PoleZero, 2-33
HPF_Polynomial, 2-34
HPF_RaisedCos, 2-35
Hybrid180, 4-23
Hybrid90, 4-21

I
IntegratorSML, 6-10
IQ_Demod_Data, 8-22
IQ_Demod_Setup, 8-24
IQ_DemodTuned, 3-8
IQ_Mod_Data, 8-26
IQ_Mod_Setup, 8-28
IQ_ModTuned, 3-9
IsolatorSML, 4-25

L
LimiterSML, 6-11
LoadPullSetup, 8-30
LogACDemod, 1-39
LogDC, 1-40
LogSuccDetect, 1-41
LogTrue, 1-42
LOS_Link, 4-27
LPF_Bessel, 2-38
LPF_Butterworth, 2-39
LPF_Chebyshev, 2-40
LPF_Elliptic, 2-42
LPF_Gaussian, 2-43
LPF_GMSK, 2-44
LPF_PoleZero, 2-46
LPF_Polynomial, 2-47
LPF_RaisedCos, 2-48

M
Mixer, 1-43
Mixer2, 1-48
MixerHBdata, 8-32
MixerHBsetup, 8-35
MixerIMT, 8-37
MixerIMT2, 8-42

N
N_StateDemod, 3-10
N_StateMod, 3-11

O
OpAmp, 1-53
OpAmpIdeal, 1-56

P
Pad, 4-29
ParallelSerial, 6-13
PhaseFreqDet, 5-5
PhaseFreqDet2, 5-9
PhaseFreqDetCP, 5-13
PhaseFreqDetTuned, 5-14
PhaseNoiseMod, 5-17
PhaseShiftSML, 4-31
PI4DQPSK_ModTuned, 3-13
PM_DemodTuned, 3-15
PM_ModTuned, 3-16
PM_UnwrapDemodTuned, 3-17
PwrSplit2, 4-33
PwrSplit3, 4-34

Q
QPSK_ModTuned, 3-18
QuantizerSML, 6-14

R
ResetSwitch, 6-16
RF_PA_CKT, 7-2
RF_RX_SML, 7-3
RF_TX_SML, 7-4

S
SampleHoldSML, 6-17
Sampler, 6-18
SAW_Filter, 2-51
SerialParallel, 6-22
SPDT_Dynamic, 6-23
SPDT_Static, 6-25
SwitchV, 6-27
SwitchV_Model, 6-29

T
TimeDelay, 4-35
Transformer, 4-36
TransformerG, 4-37
TwoPort, 4-38

V
VCA_Data, 8-48
VCA_Setup, 8-51
VCO, 5-19
VCO_DivideByN, 5-22
VMult, 1-58
VSum, 6-30