Appendix

Harmonic Balance Simulator
Harmonic Balance
for Large Signal AC and S-parameter Simulation

Harmonic Balance is a frequency domain analysis technique for simulating distortion in nonlinear circuits and systems. This method assumes the input stimulus consists of a relatively few steady-state sinusoids. Therefore, the solution can be expressed as a sum of steady-state sinusoids, that includes the input frequencies in addition to any significant harmonics or mixing terms.

\[ v(t) = \text{real} \left( \sum_{k=0}^{N} V_k e^{j \omega_k t} \right) \]

Where \( V_k \) is the complex amplitude and phase at each of a limited set of \( N \) frequencies \( \omega_k \). The circuit simulator converts nonlinear differential equations into the frequency domain, where the \( y \) become a set of nonlinear algebraic equations:

\[ j \omega_k * F_k (q(v(t))) + F_k (f(v(t))) + V_k * H(j \omega_k) = I_m(\omega_k) \]

where \( F_n() \) signifies the \( k^{th} \) spectral component of a Fourier transformation. The harmonic balance simulator then must simultaneously solve this set of \( N \) nonlinear algebraic equations for all the \( V_k \) values. The number of nonlinear equations that must be solved has increased by a factor of \( N \) relative to standard time domain simulators. This means that the matrix sizes and memory requirements of harmonic balance increase considerably as \( N \) becomes large. The nonlinear devices are still evaluated in the time domain by using an inverse Fourier transformation to convert the \( V_k \) values into the \( v(t) \) waveform prior to evaluating the nonlinear \( q() \) and \( f() \) functions. This means that standard SPICE-type, nonlinear current and charge waveforms are transformed into the frequency domain at each iteration so their spectral values can be used in the frequency domain equations. Since most harmonic balance simulators use Newton-Raphson techniques, the derivatives (nonlinear resistance and capacitance) must also be computed in the time domain and transformed into the frequency domain.

The primary advantage of harmonic balance over time domain solutions is that linear devices with arbitrary frequency responses can be easily, yet quickly, modeled. Lump ed elements approximations are no longer required. Time domain convolution has been replaced with simple frequency domain multiplication. This is especially important for RF, microwave and millimeter frequencies, which are often characterized with measured frequency domain data. An additional benefit is that harmonic balance solutions directly provide the steady-state solution without having to wait until the transient solutions dies out. For high Q circuits this can be a costly wait. The input stimulus frequencies \( \omega \) can also be arbitrarily widely spaced and may actually be non-commensurate, but the harmonic balance solution can be still quickly obtained. The complexity and cost of the solution does not increase just because there is a low frequency tone (a long period) coexisting with high frequency tones (very small time steps).
The limitations of harmonic balance are that the signal must be quasi-periodic; they must be representable as a sum of a relative few number $N$ of discrete tones. As $N$ becomes large, the amount of required internal memory becomes excessive since internal matrix sizes grow as $N^2$. This means that it is inefficient to simulate signals with arbitrary modulation, or long duration modulation, or systems with numerous LO and RF frequencies. It is practically impossible to simulate the transient response of a high frequency circuit, since the waveform must be quasi-periodic. The spectrum must consist of discrete, steady-state tones. A practical limit for today’s computers is around 64 tones for a medium size circuit.

Harmonic Balance is usually the method of choice for simulating analog RF and microwave problems, since these are most naturally handled in the frequency domain. Examples of devices and circuits suited to this analysis include power amplifiers, frequency multipliers, mixers, and modulators, under large signal sinusoidal drive. In the context of high frequency circuit and system simulation, harmonic balance has a number of advantages over conventional time-domain transient analysis:

- Designers are usually most interested in a system’s steady-state behavior. Many high frequency circuits contain long time constants that require conventional transient methods to integrate over many periods of the lowest-frequency sinusoid to reach steady state. Harmonic balance on the other hand, captures the steady-state spectral response directly.

- The applied voltage sources are typically multitone sinusoids that may have very narrow or very widely spaced frequencies. It is not uncommon for the highest frequency present in the response to be many orders of magnitude greater then the lowest frequency. Transient analysis would require integration over an enormous number of periods of the highest-frequency sinusoid. The time involved in carrying out the integration is prohibitive in many practical cases.

- At high frequencies, many linear models are best represented in the frequency domain. Simulating such elements in the time domain by means of convolution can result in problems related to accuracy, causality, or stability.

The Component Palette List provides a variety of related harmonic-balance simulation components, each in their own palettes:

- Simulation-HB – for general harmonic balance simulation.
- Simulation-LSSP – for large-signal S-parameter simulation.
- Simulation-XDB – for finding gain-compression points.

The Harmonic Balance Simulation component, in the Simulation-HB palette, obtains frequency-domain voltages and currents. Use this component to achieve the following:

- Determine the spectral content of voltages and/or currents in the circuit, and, by extension, compute quantities such as third-order intercept (TOI or IP3) points, total harmonic distortion (THD), and intermodulation distortion components (in the case of multitone stimuli).

- Perform power amplifier load-pull contour analysis.

- Perform nonlinear noise analysis.
The nonlinear **LSSP Simulation** component, in the Simulation-LSSP Palette, facilitates the computation of large-signal S-parameter in nonlinear circuits such as those that employ power amplifiers and mixers. In the latter case, the large-signal S-parameters can be computed “across frequencies” – that is, from the RF input to the IF-output.

To write simulation results that can be used by a file-based amplifier (such as the AmplifierP2D components, in the System-Amps & Mixers palette), use the P2D analysis, also in the Simulation-LSSP palette. The use of such a file can speed up subsequent simulations.

The **XDB Simulation** component, in the Simulation-XDB palette, makes it possible to determine automatically the X dB gain compression point of an amplifier or mixer. The simulator sweeps the input power upward from a small value, stopping when the required amount of gain compression is seen at the output.

The above mentioned components share a variety of options.

- **Small-signal mode** (not available in the LSSP simulator) is a small-signal/large-signal analysis that can speed up those mixer simulations where the RF input can be treated as a small-signal perturbation, while the large-signal LO components are computed by means of a full harmonic balance analysis. The small-signal/large-signal analysis can also be used in numerous other scenarios involving both large- and small-signal tones.

- **Nonlinear Noise** allows performing a nonlinear spot-noise or swept-noise simulation, and calculating noisy two-port parameters.

- **Oscillator**-Analysis is designed to analyze oscillators, including phase noise.

**How the Harmonic-Balance-Simulator Operates**

Harmonic Balance simulation makes possible the multitone simulation of circuits that exhibit intermodulation frequency conversion. This includes frequency conversion between harmonics. Not only the circuit itself produces harmonics, but also each signal source (stimulus) can produce harmonics or small-signal sidebands. The stimulus can consist of up to twelve nonharmonically related sources. The total number of frequencies in the system is limited only by such practical considerations as memory, swap space, and simulation speed. The harmonic balance method is iterative. It is based on the assumption that for a given sinusoidal excitation there exists a steady-state solution that can be approximated to satisfactory accuracy by means of a finite Fourier series. Consequently, the circuit node voltages take on a set of amplitudes and phases for all frequency components. The currents flowing from nodes into linear elements, including all distributed elements are calculated by means of straightforward frequency-domain linear analysis. Currents from nodes into nonlinear elements are calculated in the time-domain. A frequency-domain representation of all currents flowing away from all nodes is available. According to Kirchhoff’s Current Law (KCL), the currents should sum to zero at all nodes. The probability of obtaining this result on the first iteration is extremely small.
Therefore, an error function is formulated by calculating the sum of currents at all nodes. This error function is a measure of the amount by which KCL is violated and is used to adjust the voltage amplitudes and phases. If the method converges (that is, if the error function is driven to a given small value), then the resulting voltage amplitudes and phases approximate the steady-state solution.
TAHB and HBAHB

 transient Assisted Harmonic Balance: Highly nonlinear circuits with sharp-edged waveforms (such as frequency dividers) can use transient simulation results to provide an initial guess for Harmonic Balance.

The simulator automatically generates the transient initial guess when needed in Auto mode. You can also force it On or Off. Additionally, you can also set up and run an independent Transient analysis and then name the file in the Initial Guess area.

Harmonic Balance Assisted Harmonic Balance - This is turned On or Off here, but in the Auto mode, it is used when needed.

Circuits with several fundamental tones, such as mixers, will benefit from HBAHB in speed and accuracy. The simulator first does a one-tone analysis and then it uses the results to complete the analysis for the other tones and mixing products.

Convergence

Use the Auto mode unless the circuit does not converge. In that case, try the Advanced (Robust) mode, which will take more time. Increasing the Max Iterations can also help. Beginning with ADS 2005A, convergence issues and the need to use advanced methods is greatly reduced.

The Krylov Subspace Solver

Modern harmonic balance simulators rely on the Newton-Raphson technique to solve the nonlinear systems of algebraic equations that arise in large-signal frequency-domain circuit simulation problems. Each iteration of Newton-Raphson requires an inversion of the Jacobian matrix associated with the nonlinear system of equations. When the matrix is factored by direct methods, memory requirements climb as $O(H^2)$, where $H$ is the number of harmonics. Thus, the factorization of a Jacobian at $H=500$ will require 2500 times as much RAM as one at $H=10$.

An alternative approach to solve the linear system of equations associated with the Jacobian is to use a Krylov subspace iterative method such as GMRES (generalized minimum residual). This method does not require the explicit storage of the Jacobian matrix $J$, but rather only the ability to carry out matrix-vector products of the form $J^*V$, where $V$ is an arbitrary vector. It turns that the information needed to carry out such an operation can be stored in $O(H)$ memory, not $O(H^2)$, in the context of harmonic balance. Thus Krylov solver offers huge savings in memory requirements for large harmonic-balance problems. Similar arguments show that even larger increases in computational speed can be obtained.

NOTE: For all other settings in Harmonic Balance, refer to the HELP manuals for more details and information.
Fundamentals of Large-Signal S-parameter Simulation

Unlike small-signal S-parameters, which are based on a small-signal simulation of the linearized circuit, large-signal S-parameters are based on a harmonic balance simulation of the full nonlinear circuit. Since harmonic balance is a large-signal simulation technique, its solution includes nonlinear effects such as compression. This means that the large-signal S-parameter can change as power levels are varied. For this reason, large-signal S-parameters are also called power-dependent S-parameter.

Like small-signal S-parameters, large-signal S-parameters are defined as the ratio of the reflected and incident waves:

$$S_\theta = \frac{B_i}{A_j}$$

The incident and reflected waves are defined as:

$$A_j = \frac{V_j - Z_{0j}I_j}{2*\sqrt{R_0}}, \quad B_i = \frac{V_i + Z_{0i}I_i}{2*\sqrt{R_0}}$$

where

- $V_i$ and $V_j$ are the Fourier coefficients, at the fundamental frequency, of the voltages at ports $i$ and $j$.
- $I_i$ and $I_j$ are the Fourier coefficients, at the fundamental frequency, of the currents at ports $i$ and $j$.
- $Z_{0i}$ and $Z_{0j}$ are the reference impedances at ports $i$ and $j$.
- $R_0$ and $R_0$ are the real parts of $Z_{0i}$ and $Z_{0j}$.

This definition is a generalization of the small-signal S-parameter definition in that $V$ and $I$ are Fourier coefficients rather than phasors. For a linear circuit, this definition simplifies to the small-signal definition. The simulator performs the following steps to calculate the large-signal S-parameter for a two-port.

1. Terminate port 2 with the complex conjugate of its reference impedance. Apply a signal with a user-specified power level $P_1$ at port 1, using a source whose impedance equals the complex conjugate of that port’s reference impedance. Using harmonic balance, calculate the currents and voltages at port 1 and 2. Use this information to calculate $S_{11}$ and $S_{21}$.

2. Terminate port 1 with the complex conjugate of its reference impedance. Apply a signal of power $P_2 = |S_{21}|P_1$ at port 2 using a source whose impedance equals to complex conjugate of the reference impedance of port 2. Using harmonic balance calculate the currents and voltages at port 1 and 2. Use this information to calculate $S_{12}$ and $S_{22}$.
Fundamentals of Nonlinear Noise Simulation

The results of the harmonic balance simulation are output to the dataset and are then used to determine the periodic operation point for the nonlinear noise simulation. The periodic operation point is the steady-state voltage and currents within the circuit at the fundamental, harmonic, and all mixing frequencies. Every upper and lower noise sideband is modeled for each large-signal spectral component; consequently, the number of noise frequencies simulated is double the harmonic frequencies and the sparse-matrix memory requirement is quadrupled (unless the Krylov option is used).

The result is that a nonlinear noise simulation requires four times the memory of a normal harmonic balance simulation. Use low values for the parameter *Maximum order* to limit the demands on computer memory.

If computer memory is insufficient for a noise simulation, eliminate the large-signal tone at the input and replace it with a Term component. Also, reduce the number of tones in the Harmonic Balance Simulation component by 1. The results of the noise figure simulation should not change significantly.

If the noise figure is needed, add ports to the circuit. In addition, because the single sideband definition of noise figure is used, the correct input sideband frequency must be specified, by the parameter *Input frequency*. This identifies which input frequency will mix to the spot noise frequency of interest.

If the input frequency is not specified, it is assumed to be the same as the spot noise frequency. This way, the input and output frequencies are the same, as is typical of amplifiers. Input frequency is the frequency at which the signal enters the mixer RF port. For mixers, Input frequency is typically determined by an equation that involves the local oscillator frequency and the noise frequency. Either the sum or the difference between these values is used, depending on whether upconversion or downconversion is taking place. If an upper or lower sideband cannot be found that agrees with the specified Input frequency value, a warning is issued and the nearest sideband is used.

If noisy two-port parameters such as $S_{opt}$, $R_n$, or $NF_{min}$ are required, select Calculate noisy two-port parameters. Noise data will be computed only for those nodes requested. The noise voltages and currents scale with the square root of the noise parameter Bandwidth. The default bandwidth is 1 Hz, so that the results have units $\sqrt{Hz}$. Include port noise tells the simulator to include the contribution of port noise in the analysis of noise voltages and currents. Use all small-signal frequencies gives more accurate results, but also requires more memory.
Fundamentals of Oscillator Simulation

Oscillators are nonlinear by nature, using positive feedback to achieve oscillation. The loop gain must be greater than one with a phase shift of zero. Once it has been placed in the appropriate point in the circuit, the OscPort (oscillator port) component checks for this condition. When the loop-gain calculations are being performed, the oscillator port becomes a short circuit to all harmonic frequencies and a unidirectional isolator to the fundamental. During this calculation, the port effectively turns the closed-loop oscillator into an open-loop amplifier.

Although a loop gain greater than one is required for oscillation to begin, at some level the amplitude of oscillation is limited. This limit may come about through the saturation of some active device, or may be the result of an active gain control (AGC) network.

Either way, the limiting action requires that the gain of the loop be a function of the amplitude of the signal. This requires a nonlinear analysis. The simulator uses an iterative process to try to find a set of nonzero steady-state solutions.

When the oscillator simulation is launched, a harmonic balance search algorithm varies the operation frequency to find the oscillation frequency, as well as the final operating frequency and power levels, of the oscillator circuit.

The simulator performs a fully automated search for the oscillator's operating characteristics. The automated search involves three steps:

1. The search for a frequency where the circuit satisfies the linear oscillation condition.
2. The search for a frequency and power such that the oscillator's open-loop gain is at unity magnitude and zero phase angle (within a given tolerance).
3. A full closed-loop harmonic balance analysis in which frequency and oscillator power is further refined to a more exact solution.
Phase Noise Simulation

An oscillator phase noise analysis computes the noise sidebands of the oscillator carrier frequency. It also computes the following:

- Normal frequency conversion
- Amplitude-noise-to-frequency-noise conversion
- Frequency translation of noise. This is caused by component nonlinearities in the presence of large-signal oscillator signals. Upconverted flicker noise is a commonly observed effect.
- Bias changes due to the oscillator signals. Any shift in DC bias that occurs in the presence of the oscillator waveforms is taken into account. This bias-shift calculation is needed for accurate calculation of nonlinear device noise.

Selecting *Include FM noise* causes the phase noise analysis to include additional phase noise results from a second phase noise analysis based on a frequency sensitive technique. The results from the basic phase noise analysis have “PNMX” (phase noise, in $\text{dBc}/\sqrt{\text{Hz}}$) and “AMMX” (AM noise, in $\text{dBc}/\sqrt{\text{Hz}}$) appended to the selected node name(s) in the dataset. If the option is selected, an additional phase noise result is computed with “PNFM” (phase noise, in $\text{dBc}/\sqrt{\text{Hz}}$) appended to the selected node name(s).

**Note:**
Many (if not all) of the nonlinear devices in the semiconductor libraries do not have stable $1/f$ noise parameters. Instead, noise sources must be added to model the $1/f$ noise.

How ADS Simulates Phase Noise

Phase noise in an oscillator can be analyzed by two separate, independent methods:

- From the oscillators frequency’s sensitivity to noise.
- From small signal mixing of noise.

The frequency sensitivity to noise may be viewed as the oscillator acting as a VCO and changing its operating frequency due to FM modulation caused by noise generated in the oscillator.

The small-signal mixing of noise comes from the nonlinear behavior of the oscillator, where noise mixes with the oscillator signal and harmonics to mix to sideband frequencies on either side of the oscillator signal. These two models are two different ways of looking at the same problem.

The mixing analysis for the phase noise is automatically performed by Advanced Design System when an oscillator analysis is performed with nonlinear noise. The frequency sensitivity to noise must be turned on by the user by selecting *Include FM noise*. 
These two models are different views of the same process. The phase noise computed by both models is available to the users directly in dBc. Assuming there is a node labeled “vout” for which phase noise is to be computed, the following variables are available for use in a presentation following a phase noise analysis.

- `hb_noise.vout.pnfm`: phase noise (in dBc) from frequency sensitivity analysis
- `hb_noise.vout.pnmx`: phase noise (in dBc) from mixing analysis
- `hb_noise.vout.anmx`: AM noise (in dBc) from mixing analysis

Both variables `hb_noise.vout.pnmx` and `hb_noise.vout.pnfm`, which represent the different models of phase noise, should be plotted to view the simulated phase noise results. There should be a region of offset frequencies where they both produce similar results.

The frequency sensitivity to noise is obtained from the large signal (harmonic balance) oscillator solution. The sensitivity of the oscillation frequency $\omega_0$ is obtained with respect to any injected noise in the circuit. After summing over all of the noise sources, the total spectral density of frequency fluctuations is obtained and converted to phase noise. This noise has a characteristic shape of $f^{-2}$ or $f^{-3}$ if $1/f_{\text{noise}}$ sources are present.

This description is valid at small offset frequencies, but is no good at large offsets as it goes to zero; it will not exhibit a noise floor but continues down as $f^{-2}$. The ADS model for frequency sensitivity to noise is based on a sensitivity analysis (evaluated at DC) of the oscillator frequency due to perturbations from noise. This model may not predict phase noise well if there are lowpass filters in the circuit that tend to filter out noise, e.g., power supply decoupling.

To model oscillator phase noise with a noise mixing analysis, the noise at the sidebands on either side of the carrier ($\omega_0 \pm \omega$) are obtained from a small-signal mixer analysis where noise sources ($\omega \pm \omega_0$) mix with the oscillator large signal ($K\omega_0$) to produce noise sidebands. The noise at these two sideband frequencies and their correlation's is then used to compute the phase noise. The phase noise from a mixing analysis tends to be valid at large offset frequencies and will show a finite noise floor. The mixing analysis additionally computes the AM noise as well as the phase noise.