Symbolic Distortion Analysis of Analog Integrated Circuits

Wim Verhaegen* and Georges Gielen*

Abstract — A technique for generating symbolic expressions for the distortion in weakly nonlinear analog integrated circuits is presented. This technique uses some acceptable assumptions to reduce the task of analyzing the nonlinear circuit to a repeated analysis of derived linear circuits. This repetitive algorithm has been implemented and it is demonstrated on an example circuit.

1 Introduction

In the analysis of analog integrated circuits, distortion and intermodulation are important factors. Either they are unwanted, as is the case in linear building blocks like opamps, or they are explicitly wanted to obtain a signal shifted in frequency, as is the case with mixers. Distortion and intermodulation need to be assessed accurately in both cases.

Classical numerical simulation techniques using iterative algorithms for solving the differential equations are slow and inaccurate due to the large difference between the time constants normally present in the circuits of interest. Several numerical methods have been developed to overcome this problem, e.g. the harmonic balance technique [1], multitime analysis [2] and the use of describing functions in circuits with feedback [3]. However, the numerical nature of these techniques implies that no symbolic results can be derived, so that re-use of results — in the form of design equations — is not possible.

An analysis technique that does yield symbolic results is described in this paper. Based on a set of assumptions, the analysis of a weakly nonlinear circuit is reduced to a number of analyses of linear circuits. A linear symbolic analysis core is used for these individual analysis steps, and its results are combined and manipulated to get a closed-form symbolic end result. This result can be used as a design equation, or the impact of the circuit nonlinearities on distortion and intermodulation can be derived from it.

Before explaining this technique, it is to be noted that similar approaches have been followed in the past to obtain symbolic expressions for the distortion in specific classes of circuits. E.g. the distortion in sampling mixers is analyzed in [4], and a method for analyzing the distortion in analog building blocks is presented in [5]. All symbolic approaches are intrinsically limited somehow, and these publications are no exceptions. The scope of the algorithm presented in this paper is limited to weakly nonlinear circuits. This means that the circuit characteristics are nonlinear in a smooth way, implying that higher-order contributions are always smaller than lower-order ones, and that the applied signals are small.

The basic terminology used is explained in section 2, followed by a brief explanation of the algorithm in section 3. The implementation is demonstrated in section 4, and the conclusions are presented in section 5.

2 Terminology

The following terms are used in this paper:

- An n-dimensional conductance is a conductance with n controlling branch voltages. When analyzing analog integrated circuits, n equals 3 at most. The corresponding branch voltages are denoted as \( v_i, v_j \) and \( v_k \) in this paper.

- A nonlinear current is described using its DC component and the derivatives to the controlling branch voltages. The derivatives up to order 3 are described by nonlinearity coefficients. E.g. the \( q \)th-order nonlinearity coefficient \( K_{mg}^{n_1,n_2,n_3} \) of a three-dimensional conductance for \( q \geq 2 \) is given by

\[
\frac{1}{m!n!(q-m-n)!} \frac{\partial^i (v_i, v_j, v_k)}{\partial v_i^m \partial v_j^n \partial (q-m-n)^{v_k}}
\]

where \( g_1, g_2 \) and \( g_3 \) are the conductances through which the current controlled by respectively \( v_i, v_j \) and \( v_k \) flows.

- The component of a signal at a frequency which is a linear combination of the input–signal frequencies, is denoted a phasor. E.g. the phasor \( V_{out,0,1,1} \) is the component of the signal \( v_{out,0} \) at frequency \( f_{in,1} + f_{in,2} \). The signal \( v_{out,0} \) is reconstructed from all corresponding phasors using the formula

\[
v_{out,0}(t) = \frac{1}{2} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} V_{out,0,m,n} e^{j(m\omega_1+n\omega_2)t}
\]

with \( \omega_x = 2\pi f_x \).

3 Algorithm

The algorithm used for analyzing weakly nonlinear circuits is explained in [6]. Only the conclusions are given here for reasons of brevity. There is one basic formula for calculating a phasor:
\[ V_{x,m,n} = \sum_i n L I_{iNL} H_{iNL \rightarrow v_x}(mj \omega_1 + nj \omega_2) \]  

with \( i_{NL} \) a fictitious current source, and \( H_{iNL \rightarrow v_x}(s) \) a linear circuit function expressing the relation between the fictitious current and the entity of interest (in case the voltage \( v_x \)). As is shown in [6], the fictitious current source is defined by the dimensionality of the associated conductance, and the values of the coefficients \( m \) and \( n \), under the assumption that the circuit behaves in a weakly nonlinear way. E.g. the fictitious current source for a one–dimensional conductance and \( m = 2 \) and \( n = -1 \) is given by

\[ K_{2_{i-1}} \left( V_{i,1,0} V_{i,1,-1} + V_{i,0,-1} V_{i,2,0} \right) + \frac{1}{2} K_{3_{i-1}} V_{i,1,0} V_{i,0,-1} \]  

Note that equation (4) refers to phasors of order 2, which can in turn be found using equation (3). These formulas are thus applied recursively, until an expression with linear phasors only — i.e. \((m, n) = (\pm 1, 0)\) or \((m, n) = (0, \pm 1)\) — is obtained.

The linear–analysis engine Symba is subsequently used to replace the linear transfer functions in the resulting expression with functions of the (linear) circuit elements. To this purpose, Symba internally uses a numerical approximation algorithm [7], and a matroid–based symbolic term generator [8] combined with an error controller.

The resulting expression with the linear circuit elements can grow quite large for circuits of low complexity already, but luckily many terms in it are negligible. The expression is therefore further simplified using a number of sorting and elimination techniques like numerical screening of the terms and adaptive control of the approximating linear analysis, controlled with the appropriate error algorithms. This finally results in an expression of the form (3), with a reduced number of terms.

Note, finally, that equation (3) yields an harmonic of a current or voltage. Many distortion specs — e.g. OIP2, IP3 — are expressed as a combination of several such components. For these cases we must repeat the nonlinear analysis process for the different components and combine the end results. An example of this will be given in section 4.

The repetitive application of equation (3) using nonlinearity stamps like the one in equation (4) has been automated in a nonlinear–analysis engine, which interacts with the user and the linear–analysis engine Symba. This engine has been implemented in Perl and C++ and is linked to the Symba linear–analysis engine. The use of the nonlinear–analysis engine on an example circuit is demonstrated in section 4.

4 Applications

The distortion in the Miller opamp shown in figure 1 is now analyzed as an illustration of the use of the nonlinear–analysis engine. The opamp has a differential input at the matched source–coupled pair M1a and M1b, and a single–ended output. The unity feedback obtained with the resistor Rfb configures the opamp as a voltage follower. The opamp has a gain–bandwidth product of 1 MHz, and we are interested in the second–order distortion below that frequency.

The application of all the fictitious current sources results in a second–order response at the output \( V_{\text{out},0},2 \) which is the product of \( \frac{1}{2} \omega \) and the sum of contributions shown in table 1. The nonlinear coefficients \( K_i \) in the contributions are calculated as the derivatives of a transistor characteristic fitted through points obtained using a numerical simulator.

In order to interpret the expression for \( V_{\text{out},0},2 \) and obtain an expression for, say, the second–order output–referred intermodulation product OIP2, the contributions (5) to (19) have to be further simplified. This is achieved in 4 subsequent steps:

1. The entries that contribute only a little to the value of \( V_{\text{out},0},2 \) can be eliminated with a small loss of accuracy. This elimination needs to be validated over the entire frequency range of interest. In order to facilitate this, a relative weight is assigned to each entry using the following formula:

\[ w_i(f) = \frac{|e_i(j \omega)|}{|V_{\text{out},0},2(j \omega)|} \]  

where \( e_i(f) \) is the \( i \)’th entry as a function of the frequency. These weights are calculated based on the numerical reference value of the different terms, without any symbolic calculations.

Based on these weights, the entries (5) to (19) are examined for elimination in order of increasing maximum weight. The impact of the entries
2. The transfer functions found in the contributions (5) to (15) are paired, with
coefficients are (approximately) equal. The differences in the total distortion expression, it is found
that all contributions (5) to (15) are paired, with
to entry (11) anyway.

All these substitutions and eliminations finally result
in the symbolic formula:

\[
V_{\text{out},2} = \frac{V_{\text{in}}^2}{2} \left( \frac{K_{22m}M_3}{8M_3} + \frac{K_{22m}M_1 - K_{22m}M_4}{8M_1} \right)
\]  

(21)

4. Finally the output–referred second–order intercept point \( OIP_2 \) is calculated, which is defined as the
input amplitude for which \( |V_{\text{out}}| = |V_{\text{out},2}| \), i.e.

\[
V_{\text{out}} = \frac{V_{\text{in}}^2}{2} \left( \frac{-8M_1 K_{22m}M_3 + 8M_3 \left( K_{22m}M_1 - K_{22m}M_4 \right)}{8M_1 M_3} \right)
\]  

(22)

So \( OIP_2 \) is given by

\[
OIP_2 = \frac{2M_1 M_3}{-8M_1 K_{22m}M_3 + 8M_3 \left( K_{22m}M_1 - K_{22m}M_4 \right)}
\]  

(23)

For the design point chosen for the Miller opamp

\[
2M_1 M_3 = 824.5 \times 10^{-9}
\]  

(24)

\[
-8M_1 K_{22m}M_3 = 424.11 \times 10^{-9}
\]  

(25)

\[
8M_3 \left( K_{22m}M_1 - K_{22m}M_4 \right) = -7.5656 \times 10^{-9}
\]  

(26)

As a result the contribution of \( K_{22m}M_3 \) in the denominator of (23) is dominant, and the expression
can be simplified to:

\[
OIP_2 = \frac{2K_{22m}M_3}{K_{22m}M_1}
\]  

(27)
which equals 1.944V for the chosen design point.

Note that the obtained results are only valid in the case of perfectly matched transistor characteristics for the pairs M1a and M1b, resp. M2a and M2b. When this assumption cannot be made, the entire exercise needs to be repeated for the same circuit with mismatch introduced. This will of course result in a lower $OIP_2$.

5 Conclusions

The extension of linear analysis to obtain closed–form symbolic expressions for the weakly nonlinear characteristics of analog integrated circuits has been demonstrated with an example circuit. The underlying algorithm and simplification of the lengthy results has been automated and linked to the existing Symba environment. The resulting simplified expressions can be used as design equations in an analog design framework.

Acknowledgments

This research has been funded in part by the Esprit project 21812 Amadeus.

References


