Symbolic Methods in Industrial Analog Circuit Design

T. Halfmann and T. Wichmann

Fraunhofer Institute for Industrial Mathematics, 67663 Kaiserslautern, Germany,
thomas.halfmann@itwm.fraunhofer.de

Abstract Industrial analog circuits are usually designed using numerical simulation tools. To obtain a deeper circuit understanding, symbolic analysis techniques can additionally be applied. Approximation methods which reduce the complexity of symbolic expressions are needed in order to handle industrial-sized problems. This paper describes aspects of the field of symbolic analog circuit analysis. Some state-of-the-art simplification algorithms for linear and nonlinear circuits are presented. The basic ideas behind the different techniques are described and two application examples for the linear and nonlinear case will be demonstrated.

1 Introduction to Symbolic Circuit Analysis

The motivation for applying symbolic techniques to the field of analog circuit design has been to gain insight into circuit behavior by interpreting analytic formulas instead of using traditional numerical design and simulation tools which lack in providing deeper design understanding. However, it becomes apparent quite quickly that exact symbolic analysis yields expressions which are too complex to be of any use. Obviously, for industrial circuits with more than just one transistor it is impossible to obtain useful results due to the extreme computational complexity of symbolic calculations. This contradicts the initial intention of symbolic analysis, namely to gain insight into unknown circuit behavior. This motivated the development of simplification algorithms which lead to a breakthrough in the field of symbolic circuit analysis.

The equation system describing the behavior of an analog circuit consists of equations originating from Kirchhoff's current and voltage laws as well as of the circuit element characteristics. It can be set up automatically using standard formulation methods such as the Modified Nodal Analysis or the Sparse Tableau Analysis. In general, the circuit equations are given by a differential-algebraic equation system (DAE system)

$$F = \begin{pmatrix} f(x(t), x'(t), y(t), u(t)\,; p) \\ g(x(t), y(t), u(t)\,; p) \end{pmatrix} = 0 \quad \text{for all } t \in I \ . \tag{1}$$

Here, $u : \mathbb{R} \to \mathbb{R}^r$ denotes the inputs, $x = (v, i) : \mathbb{R} \to \mathbb{R}^k$ denotes the vector of dependent variables, $y : \mathbb{R} \to \mathbb{R}^s$ denotes the outputs, and $I \subset \mathbb{R}$ denotes a time interval. Since we are working with symbolic equations, F is parameterized by symbolic element parameters $p = (p_1, \ldots, p_N)$ (like a resistor value R_1, a voltage source value V_0, or a transistor parameter β_F).

1.1 Symbolic Simplification Algorithms

As mentioned above, symbolic analysis of large analog circuits seems to be senseless as long as the complexity problem has not been solved. Thus, in order to reduce the complexity of the symbolic expression, one needs to simplify it.

In general, the term *symbolic simplification* or *symbolic approximation* refers to a whole family of hybrid symbolic/numeric algorithms for expression simplification. These techniques require more numerical knowledge about the investigated circuit than manual simplifications but yield compact expressions with predictable error in a fully automated way. In manual circuit analysis the decisions on which expressions to keep and which ones to discard are based on vague and only qualitative assumptions (e.g. $R_1 \ll R_2$) that do not allow for assigning precise error figures to simplified expressions. For automating the designer's behavior within a computer program one needs exact figures to simplify an expression because qualitative relations between elements are not sufficient for determining the importance of a term especially if the expression to be simplified consists of non-trivial combinations of symbols.

The basic idea behind the simplification algorithms can be outlined as follows: starting with a symbolic equation system F describing the circuit's behavior, the user chooses one or more numerical reference solutions f_i as well as an error bound ε. The algorithms then apply symbolic simplifications to the system (e.g. the deletion of an entire expression in a sum) and solve this simplified system numerically. The hereby obtained solutions $\tilde{f}_i$ are compared to the reference solutions using an appropriate error norm: $\delta_i = \|f_i - \tilde{f}_i\|$. If the error bound is exceeded, i.e. $\max \delta_i > \varepsilon$, the simplification is undone. This is repeated until no more simplifications are possible without a violation of the error bound and the simplified symbolic system $\widetilde{F}$ is returned.

The simplification algorithms assure that the numerical behavior (with respect to the chosen references f_i) of the simplified system coincides with that of the original system within the user-given error bound. Depending on the analysis task, the reference solutions f_i can for example be a numerical transfer function, its poles and zeros, or a time-dependent solution.

1.2 Ranking Methods

The order in which to simplify terms from the equation system is one of the crucial points: It is quite clear that those terms should be simplified first which have only a minor influence on the output behavior. Terms with a large influence should not be removed at all. To achieve a maximum number of simplifications and to avoid unnecessary modifications an optimized order, the so called *ranking*, should be used. For this, a ranking algorithm is needed which predicts the influence on the output a modification would cause. As the number of possible simplifications is very large it is inconvenient to exactly compute the influence and therefore estimation methods have to be used. The design of a good ranking algorithm is a trade off between accurate error prediction and computational effort.

2 Linear Symbolic Analysis

The transfer function is the main object of interest in linear symbolic analysis. It allows for obtaining insights into the circuit's behavior and parameter dependencies. By post-processing the transfer function one can for example symbolically compute its poles and zeros to investigate the circuit stability. The research on this topic started in the early 1990's (e.g. [GS91], [Som94]).

2.1 Linear Simplification Techniques

Basically, one distinguishes three types of linear simplification methods: Simplification Before Generation methods (SBG) simplify the matrix equations before computing the transfer function. Simplification During Generation methods (SDG) apply simplifications during the process of transfer function calculation. Simplification After Generation methods (SAG) simplify the transfer function directly. Now, we will describe SAG and SBG methods only.

Simplification After Generation. This technique [GS91] is based on the manipulation of the symbolic transfer function given as a rational expression

$$H(s\,;p) = \sum a_i(p)\, s^i \;/\; \sum b_i(p)\, s^i \quad , \tag{2}$$

where the coefficients $a_i = \sum a_{ij}$ and $b_i = \sum b_{ij}$ are symbolic functions of the parameter vector p given in canonical sum-of-products form. For a given error bound, those terms a_{ij} and b_{ij} are removed from the transfer function which cause a negligible deviation on a_i and b_i, respectively. By this, one can drastically reduce the symbolic complexity of the transfer function.

Simplification Before Generation. Even for circuits of small size it is not possible to calculate the full symbolic transfer function (2). For example, the $\mu A741$ operational amplifier yields a transfer function whose expanded denominator consists of more than 10^{34} terms [Hen00]. Thus, the linear equation system itself has to be simplified before computing the symbolic transfer function. This can be done by rewriting each entry of the system matrix in sum-of-products form and sequentially removing terms from the matrix. The error is checked by computing the magnitude and phase of the (numerical) transfer function at certain frequency points. SBG methods reduce both the complexity of the transfer function as well as its polynomial order. For SBG techniques a dedicated ranking method has been developed [Hen00] which makes use of the Sherman-Morrison formula.

Poles and Zeros. The extraction of symbolic expressions for poles and zeros is rarely possible without simplifications. In [Hen00], a matrix-based SBG method for direct approximation of a linear system with respect to a selected eigenvalue of a generalized eigenvalue problem was presented. By means of eigenvalue sensitivity the symbolic parameters with negligible influence on the eigenvalue are discarded from the linear system resulting in a simplified generalized eigenvalue problem whose determinant yields a reduced-order approximation of the characteristic polynomial. To detect potentially false eigenvalue pairings during approximation, the modal assurance criterion (MAC) [FM95] is applied, which constitutes a measure for the correlation of two eigenvectors u_1 and u_2 and which is defined as

$$\mathrm{MAC}(u_1, u_2) = \frac{|u_1^H u_2|^2}{(u_1^H u_1)(u_2^H u_2)} \quad . \tag{3}$$

The value of the MAC ranges from 0 (orthogonal vectors) to 1 (parallel vectors). Hence, the MAC must be very close to 1 for considering a valid approximation. Since in this context one is interested in a single eigenvalue only, it is appropriate to use an iterative generalized eigenvalue problem solver like the Jacobi orthogonal correction method [SBFV96] instead of the QZ algorithm. As an additional benefit, the MAC can be integrated within the Jacobi correction iteration. This results in a very efficient and reliable approximation method for the extraction of approximated symbolic poles and zeros.

2.2 Industrial Application

The application of the pole/zero extraction algorithm will be demonstrated on the CMOS folded-cascode operational amplifier shown in Fig. 1. The frequency response of the operational amplifier's open-loop differential-mode voltage gain (solid curve) shows a peak near 10 MHz, caused by a parasitic complex pole pair close to the imaginary axis. The analysis task is to extract a symbolic expression for the parasitic pole pair which allows to determine those circuit parameters which have a dominant influence on the peak.

Using a SPICE Level 3 AC model for the MOS devices [GM93] yields a system of 29 equations. The differential-mode voltage transfer function has 19 poles and 19 zeros and contains more than 5×10^{19} product terms. The symbolic approximation routines are applied to extract the parasitic pole pair at $s_p = (-2.1 \pm 8.3j) \times 10^7$ using a relative error bound $\varepsilon = 0.1$. The resulting simplified equation system can be algebraically reduced to a system of dimension 4 from which the wanted pole pair $s_p^{1,2}$ can be easily computed to the expression shown in Fig. 2. The overall computation time (including netlist import and

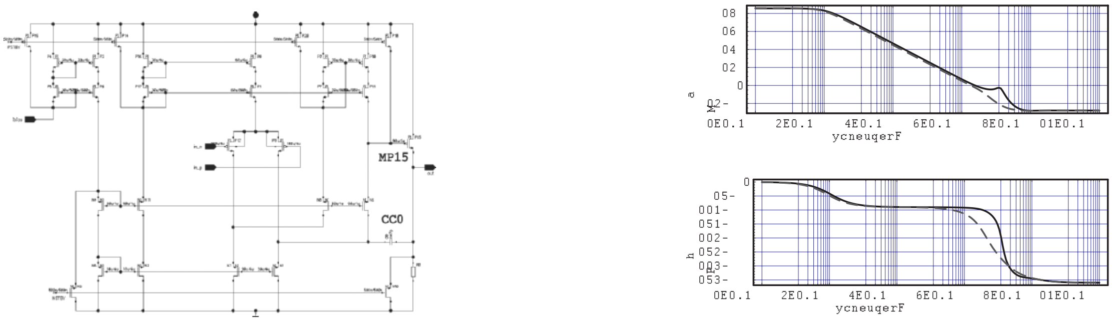

Fig. 1. CMOS folded-cascode operational amplifier

$$\mathrm{s_p}^{1,2} == -\frac{\mathrm{(CC0+CL)\ gm\$MN6}}{\mathrm{2\ CC0\ CL}} \pm \frac{\sqrt{\mathrm{Cgs\$MP15 gm\$MN6 (Cgs\$MP15\ (CC0+CL)^2\ gm\$MN6 - 4\ CC0^2 CL gm\$MP15)}}}{\mathrm{2\ CC0\ Cgs\$MP15 CL}}$$

Fig. 2. Computed formula for the complex pole pair

equation setup) to approximate the equation system and to extract this formula is about 8 seconds running the routines under Mathematica 4.0 on an AMD Athlon 1200 with 512 MB memory. By interpretation of the computed formula for the complex pole pair it turns out that an increased value for the gate-source capacitance `Cgs$MP15` of the transistor MP15 allows for decreasing the imaginary parts of the pole pair. As a consequence one could add an additional capacitor between the gate and the source terminals of the corresponding transistor and by that remove the peak in the transfer function of the voltage gain of the operational amplifier (dashed curve).

3 Nonlinear Symbolic Analysis

Simplification methods for linear analog circuits have been successfully applied to industrial applications for several years. In the following, we want to describe how the presented methods can be extended to the analysis of nonlinear circuits. Research on this topic started a few years ago and is still in progress (e.g. [Bor98], [PHHB98], [WPHH99], [Wic01]).

3.1 Nonlinear Simplification Techniques

As opposed to the linear case, for nonlinear circuits in general we can not expect to obtain explicit formulas for the solution of the output variables. In contrast to linear symbolic analysis, nonlinear simplification techniques are therefore mainly used for automated behavioral model generation. However, for small circuits it can be possible to obtain an interpretable result.

Behavioral model generation is a technique for speeding up numerical simulation of large circuits. The idea is to replace each frequently used subblock in the circuit by a single simplified model description and by that reducing the complexity of the whole circuit and decreasing the computation time. Nonlinear simplifications can be used to automatically generate behavioral models from netlist descriptions of the subblocks [NHB02].

The general outline of the nonlinear simplification algorithms is described in Sect. 1.1. The following simplification techniques have been developed to reduce the complexity of nonlinear DAE systems:

- *Elimination*: Solving equations explicitly for one variable and substituting this variable in the remaining system, thus reducing the number of equations.
- *Simplification of piecewise-defined functions*: Detecting and removing branches of piecewise-defined functions which are unused for the given input-value set.
- *Deletion of terms*: Omitting terms of sums which do not participate a significant part to the sum.
- *Substitution of terms by constant values*: Substituting a constant value for terms which do not participate a significant part to the sum.

3.2 Operational Amplifier Example

Figure 3 shows the schematic of an operational amplifier consisting of eight bipolar transistors. The input signal is given by the pulse wave voltage source VSig, the node voltage v_9 at node 9 is the output signal (dashed curve). Using the Gummel-Poon equations for modeling the bipolar transistors, the transient behavior of the operational amplifier can be described by an equation system consisting of 73 equations and variables. Nonlinear symbolic simplfications techniques are now used for generating a behavioral model of the circuit. For this, a maximum transient error of 1 V for the output variable is chosen (gray shaded area).

To automatically reduce the complexity of the equation system, symbolic simplification techniques are applied in the following order: Elimination, simplification of piecewise-defined function, cancellation of

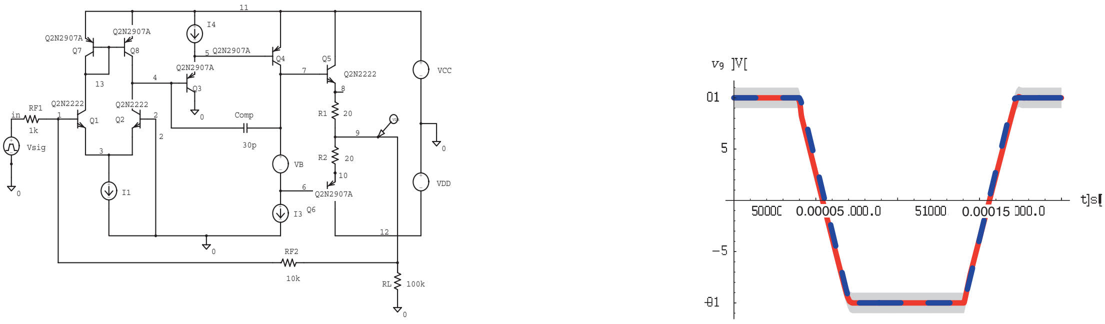

Fig. 3. Bipolar operational amplifier

terms, and again elimination. This results in a system of 6 equations only with an overall computation time of 792 s. The numerical solution of the simplifed system (solid curve) lies within the specified error margin. Solving the original DAE system numerically on the time interval $t \in [0\,\mathrm{s}, 0.002\,\mathrm{s}]$ takes about 166 s, whereas solving the reduced equation system takes about 2.3 s. Thus, the generated behavioral model yields a simulation speed-up of more than a factor of 70.

3.3 Nonlinear Ranking Methods

In principle, nonlinear ranking methods have to be designed for each simplification method and each analysis method separately. In the following we will briefly describe the *one-step solver ranking*, a ranking method which measures the influence of term cancellations on the transient behaviour. Additional ranking methods can for example be found in [PHHB98, WPHH99].

Let F denote the original DAE system with transient solution x_F, let G denote the simplified system, and let G_S denote the static system which results from G by replacing differentials by finite difference expressions according to the chosen integration scheme. An estimation $\tilde{x}_G$ of the (unknown) solution x_G of G is computed as follows: At each time instance t_i a Newton iteration to solve G_S is started for the initial value $x_F(t_i)$. The first Newston step

$$\tilde{x}_G^{[1]}(t_i) = x_F(t_i) - J_{G_\mathrm{S}}(x_F(t_i))^{-1} G_\mathrm{S}(x_F(t_i)) \ , \tag{4}$$

is then used as an estimation for the true solution $x_G(t_i)$. Finally, the obtained values $\tilde{x}_G^{[1]}(t_i)$ are interpolated yielding the estimation $\tilde{x}_G$ for x_G. The ranking value is then given by $\delta_G = \|\tilde{x}_G - x_F\|$. In our applications this ranking method yields very accurate error estimates with moderate computational effort.

3.4 Index Calculation

The index plays an important role in the theory of DAE systems [BCP89]. It is well known that numerical solving of systems with an index higher than 1 is an ill-posed problem. Since during nonlinear symbolic simplifications the index may increase, we want to compute the index in order to avoid index changes. For this, from the wide variety of different index concepts we have chosen to control the *tractability index* [GM86] and the *strangeness index* [KM98] during simplification. They are both defined for general nonlinear DAE systems and can be computed numerically. In our applications it turned out that the singular value decomposition yields the best numerical results for computing both the tractability index and the strangeness index. The Gram-Schmidt orthonormalization can also be used to calculate the tractability index symbolically, but the resulting expressions are too complex even for small circuits. For the operational amplifier example in Sect. 3.2 a number of 11 simplifications out of 254 had to be rejected due to an invalid increase of the index of the simplified system.

4 Conclusions

We have provided an insight into the area of symbolic techniques for the analysis and design of analog circuits. It was motivated that due to the complexity problem simplification methods are indispensable for handling industrial-sized problems. The basic ideas behind these simplification methods – a combination of symbolic and numeric algorithms – have been shown. The described techniques are integrated in the software *Analog Insydes* ([HHTW01], `www.analog-insydes.de`) which is an add-on package to the computer-algebra system Mathematica for the analysis, modeling, sizing, and optimization of linear and nonlinear circuits of industrial size.

During several years of application the symbolic simplification algorithms have proven to be applicable to industrial-sized problems and by that making symbolic analysis a powerful technique in industrial analog circuit design.

References

[BCP89] Brenan, K.E., Campbell, S.L., Petzold, L.R.: The Numerical Solution of Initial Value Problems in Ordinary Differential-Algebraic Equations. North Holland Publishing Co. (1989)

[Bor98] Borchers, C.: The symbolic behavioral model generation of nonlinear analog circuits. IEEE Transactions on Circuits and Systems II **45:10**, 1362–1371 (1998)

[FM95] Friswell, M.I., Mottershead, I.E.: Finite Element Model Updating in Structural Mechanics. Kluwer Academic Publishers, Dordrecht (1995)

[GM86] Griepentrog, E., März, R.: Differential-Algebraic Equations and their Numerical Treatment. BSB Teubner (1986)

[GM93] Gray, P.R., Meyer, R.G.: Analysis and Design of Analog Integrated Circuits, 3rd edition. John Wiley & Sons, Inc. (1993)

[GS91] Gielen, G., Sansen, W.: Symbolic Analysis for Automated Design of Analog Integrated Circuits. Kluwer Academic Publishers, Boston (1991)

[Hen00] Hennig, E.: Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits. Shaker Verlag, Aachen (2000)

[HHTW01] Halfmann, T., Hennig, E., Thole, M., Wichmann, T.: Analog Insydes 2 Manual. Fraunhofer ITWM, Kaiserslautern, Germany (2001)

[KM98] Kunkel, P., Mehrmann, V.: Regular solutions of nonlinear differential-algebraic equations and their numerical treatment. Numerische Mathematik **79:4**, 581–600 (1998)

[NHB02] Näthke, L., Hedrich, L., Barke, E.: Betrachtungen zur Simulationsgeschwindigkeit von Verhaltensmodellen nichtlinearer integrierter Analogschaltungen. In Analog 2002, Bremen, Germany, 107–112 (2002)

[PHHB98] Popp, R., Hartong, W., Hedrich, L., Barke, E.: Error estimation on symbolic behavioral models of nonlinear analog circuits. In SMACD 1998, Kaiserslautern, Germany, 223–226 (1998)

[SBFV96] Sleijpen, G.L., Booten, A.G.L., Fokkema, D.R., Van der Vorst, H.A.: Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems: Part I. BIT **36:3**, 595–633 (1996)

[Som94] Sommer, R.: Konzepte und Verfahren für den rechnergestützten Entwurf von Analogschaltungen. VDI-Verlag, Düsseldorf, Germany (1994)

[Wic01] Wichmann, T.: Simplification of nonlinear DAE systems with index tracking. In ECCTD'01. volume II, Espoo, Finland, 173–176 (2001)

[WPHH99] Wichmann, T., Popp, R., Hartong, W., Hedrich, L.: On the Simplification of Nonlinear DAE Systems in Analog Circuit Design. In: CASC 99. Springer Verlag, Berlin, Germany (1999)