Combination of Symbolic and Interval-Numeric Methods for Analysis of Analog Circuits

Alexander Dreyer*

Abstract – **Numerical simulations of analog circuits can be used to get a first impression of a circuits' behavior. But actual circuit properties may differ from the results obtained by floating-point simulations, due to errors caused by rounding, component tolerances, and simplified models. Simulations based on interval arithmetic can be used as a unified framework to bound all these errors, but tend to be too conservative. In this paper a combined symbolic/interval-numeric approach, tuned to treat AC and DC analysis problems with uncertain parameters, is introduced.**

1 MOTIVATION

The behavior of analog circuits can be described by a system of parameter-dependent linear or nonlinear equations. A symbolic setup of the equation system allows for assigning unique symbols to each circuit component parameter. The resulting circuit equations cannot be analyzed in a pure symbolic way: In the nonlinear case the system might not be solvable in a symbolic manner, but yet in the linear case the result may be of large complexity. In order to analyze the behavior of such an analog circuit using a simulator or a numerical solver, the symbols representing netlist elements have to be replaced by the corresponding numerical values, according to a given design point.

The numerical approach has two major drawbacks: First of all, for an efficient numerical treatment of the equation system, all numerical values have to be converted into floating-point numbers. This may lead to an excessive growing of the overall error, due to rounding of the numerical values in every solving step. Another problem is caused by the fact that a design point may be defined in advance, but one cannot ensure a priori that the desired properties will exactly be met during manufacturing of the actual circuit. Component tolerances will always lead to small variations of a circuit's properties, which may result in effects not expected from the results of the numerical simulation. While rounding errors could be reduced or even completely avoided by a sophisticated treatment of the equation system, the latter problem cannot be overcome within a single numerical simulation. Simulation based on *interval arithmetic* can be used as a unified framework for both problems.

2 INTERVAL ARITHMETIC

2.1 Basics

If upper and lower bounds for the uncertain parameters can be determined, these can be interpreted as the endpoints $\underline{x}, \overline{x}$ of a closed interval $[\underline{x}, \overline{x}] \subseteq \mathbb{R}$. This interval is usually denoted by $[x]$. A vector of intervals – or *box* – is consequently written as $[\mathbf{x}]$. The principles of interval arithmetic are quite simple (e.g. [6, 7]): during evaluation any expression is constructed by subsequent calls of elementary binary operations (say: $+, -, *, /$) and basic functions like *sin*, *cos*, *log*, e^x and x^n. The *intervalization* of these expression elements is easy to obtain, e.g.:

$$\begin{aligned} &[x] \cdot [y] = [\underline{z}, \overline{z}], \text{ for } \cdot \in \{+, -, *, /\}, \\ &\text{with } \underline{z} = \min\{\underline{x}\cdot\underline{y}, \underline{x}\cdot\overline{y}, \overline{x}\cdot\underline{y}, \overline{x}\cdot\overline{y}\}, \\ &\text{and } \overline{z} = \max\{\underline{x}\cdot\underline{y}, \underline{x}\cdot\overline{y}, \overline{x}\cdot\underline{y}, \overline{x}\cdot\overline{y}\}. \end{aligned} \tag{1}$$

Functions like $e^{[x]}$ and $[x]^n$ can be defined in an analogous manner. The intervalization of any monotonic or piecewise monotonic elementary function is computed by evaluating the function on a finite set of *special points*, consisting of the interval's endpoints and local extrema.

For bounding the range of a more complex expression we have to assign a corresponding *interval extension* to it. An interval-valued function $[f]$ is called an interval extension of the real-valued function f, if

$$[f]([x_1], \ldots, [x_n]) \supseteq \{ f(y_1, \ldots, y_n) \mid y_i \in [x_i] \}. \tag{2}$$

The interval extension obtained by replacing real operations and elementary functions by their interval-valued equivalents is called *natural interval extension* [6].

2.2 Implementation Issues

In the spirit of interval computations, the result has to give lower and upper bounds to the actual range of a function. Hence implementations based on floating-point numbers have to obeye that computations are correct only up to one *unit in the last place* (*ulp*). Outer approximations can be implemented efficiently by switching the CPU's floating-point rounding mode to *round-down* or *round-up* during the computation of the infimum or supremum, respectively. Alternatively, this may

* Fraunhofer Institute for Industrial Mathematics (ITWM), Gottlieb-Daimler-Straße, D-67663 Kaiserslautern, Germany. E-mail: alexander.dreyer@itwm.fraunhofer.de

be emulated by adding the interval $[-\varepsilon_1, \varepsilon_2]$, where ε_i denote the corresponding *ulp*s. The latter will result in slightly larger intervals.

2.3 The Dependence Problem

The major drawback in using interval computations is caused by the *dependence problem*. While computations using independent parameters will return tight bounds to the exact range of the function, two or more occurrences of the same parameter during the evaluation phase will result in too conservative estimations. For instance, the expression $f(x) = x - x$ immediately evaluates to zero, using either floating-point or symbolic techniques. According to equation (1) interval-valued substraction is defined as:

$$[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}].$$

Hence $[x] - [x]$ equals zero if and only if $\underline{x} = \overline{x}$. Simply replacing numerical solvers by an interval version would soon lead to rather useless results, consisting of large parts of the original search region.

Several algorithms have been developed to solve linear and nonlinear interval-valued equation systems [6, 7]. These behave well if we are dealing with intervals of small width and little dependence of the interval-valued terms of the expressions involved, but real-life applications will require the treatment of wider intervals and parameters of multiple occurrences. For this purpose interval algorithms have to be tuned for solving of equation systems resulting from analog circuits.

3 SYMBOLIC PREPROCESSING

Symbolic preprocessing and modeling can be used to resolve and reduce dependencies.

Analysis of linear or nonlinear circuit equations leads to a sequence of large equation systems to be solved. From the computational point of view interval solvers are expensive and should be avoided whenever possible. It is useful to compress the equation system by eliminating pairs of variables and equations, which will not generate additional dependent entries. Analogously to the case of sparse real systems, where Markowitz-heuristic is used to preserve sparseness [2], the maximal number of newly generated (dependent) interval-valued terms can be bounded, and hence dependence growing can be tracked. In order to reduce the system even more, very small parameter tolerances may be treated like exact values.

Furthermore a large number of simplifications, which possibly reduce dependencies, can be applied to any expression. In particular this is based on matching and appropriate rewriting of recurring patterns from circuit

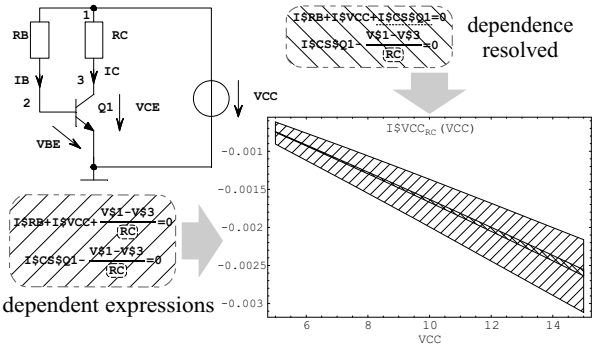


Figure 1: Influence of the equation formulation

design like:

$$\frac{R_1 R_2}{R_1 + R_2} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}. \tag{3}$$

The dependence problem may also result from parameters appearing in two or more of the circuit equations in a given formulation. For instance, see the circuit given in Fig. 1. In order to analyze its DC behavior with respect to a tolerance affected resistor $R_C \in [3.6\,\mathrm{k\Omega}, 4.4\,\mathrm{k\Omega}]$, the input voltage is swept from 5 Volts to 15 Volts (for constant $R_B = 600\,\mathrm{k\Omega}$). The outermost curves in the graph correspond to lower and upper bounds computed using interval simulations on a set of equations, in which R_C occurs twice. In contrary the inner curves show the exact range, which can easily be obtained from an equivalent system, where the second appearance of R_C was eliminated.

4 LINEAR CIRCUITS

In the case of linear analog circuits with uncertain parameters Kirchhoff's laws and element relations are summarized in a matrix equation of the following form:

$$\mathbf{A}(\mathbf{p}) \cdot \mathbf{x} = \mathbf{b}, \tag{4}$$

where $\mathbf{x}$ denotes the vector of internal currents and voltages, and $\mathbf{p} = (p_1, \ldots, p_{n_p})$ corresponds to tolerance affected components, which are bounded by intervals $[p_i]$. Solving such an interval equation system means determining close bounds to the smallest box $[\mathbf{x}]$ with

$$[\mathbf{x}] \supseteq \left\{ (\mathbf{A}(\mathbf{p}))^{-1} \cdot \mathbf{b} \mid p_i \in [p_i] \right\}. \tag{5}$$

Note that uncertain values of independent current and voltage sources can also be modeled as interval-valued parameters on the right-hand side $\mathbf{b}$. Due to their linear relationship these kind of parameters are easy to handle and hence do not need special treatment.

In order to apply interval arithmetic in linear circuit analysis, it is necessary to use a real formulation of

the matrix equation. Hence a complex-valued equation system used for AC analysis needs to be reformulated [8]. The result corresponding to each variable will be wrapped by a polygon in the complex plane (*wrapping effect*).

Earlier efforts to solve interval-valued linear circuit equations were restricted to formulations, in which each matrix element varies independently [8]. Therefore a new approach was developed to cope with multiple occurrences of parameters: First of all we will assume that the parameter dependence of $\mathbf{A}(\mathbf{p})$ can be written as a sequence of rank-one updates of a parameter-independent matrix $\mathbf{A}_0 \in \mathbb{R}^{n\times n}$:

$$\mathbf{A}(\mathbf{p}) = \mathbf{A}_0 + \sum_{i=1}^{n_\mathbf{p}} p_i \cdot (\mathbf{u}_i \cdot \mathbf{v}_i{}^T), \qquad (6)$$

with $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^n$, and $\mathbf{A}(\mathbf{p})$ is invertible for all $\mathbf{p} \in [\mathbf{p}]$. This is not a restriction at all, because this structure is already inherent to a linear circuit: Using the *sparse tableau formulation* (*STA*) [9] to generate the linear circuit equations, each p_i will occur only once and $\mathbf{u}_i, \mathbf{v}_i$ are just unit vectors, which define the corresponding matrix element. In the case of *modified nodal analysis* (*MNA*) [9] the matrices $p_i \cdot (\mathbf{u}_i \cdot \mathbf{v}_i{}^T)$ correspond to the well-known fill-in patterns used during equation setup.

Under these conditions the dependence of the solution of the linear system is given by[1]:

$$(\mathbf{A}(\mathbf{p}))^{-1} \cdot \mathbf{b} = \tilde{\mathbf{a}} - \left(\frac{1}{p_j - p_{0j}} + \tilde{c}\right)^{-1} \cdot \tilde{\mathbf{b}}, \qquad (7)$$

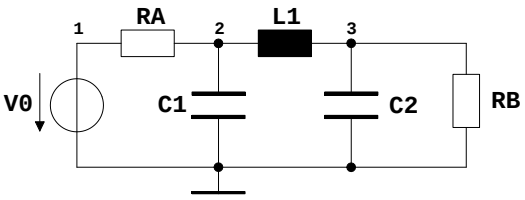
for some $\tilde{\mathbf{a}}, \tilde{\mathbf{b}} \in \mathbb{R}^n$, $\tilde{c} \in \mathbb{R}$, and $p_{0j} \in [p_j]$; each of which not depending on p_i (but possibly on some other parameter p_j). Hence $(\mathbf{A}(\mathbf{p}))^{-1} \cdot \mathbf{b}$ is monotonic and continous in each parameter p_i and the following holds:

$$[\mathbf{x}] = [\{(\mathbf{A}(\mathbf{p}))^{-1} \cdot \mathbf{b} | p_i \in \{\underline{p_i}, \overline{p_i}\}\}]. \qquad (8)$$

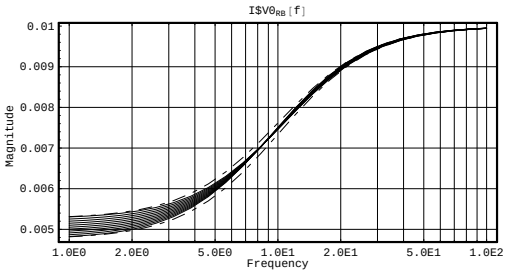
The right-hand side refers to the smallest interval vector containing the set in brackets. Hence the problem can be reformulated in terms of 2^{n_p} parameter sweeps with respect to the corners of $[\mathbf{p}]$ on a real linear system, which – for efficiency reasons – can be solved using faster floating-point solvers, in case rounding errors do not have to be tracked.

For example this technique is applied to the RLC filter shown in Fig. 2a. The interval AC analysis for the design point $V_0 = 10\,\mathrm{V}$, $R_a = 1\,\mathrm{k\Omega}$, $C_1 = 4.7\,\mu\mathrm{F}$, $C_2 = 0.22\,\mu\mathrm{F}$, $L_1 = 1\,\mathrm{mH}$ and the uncertain resistor $R_b \in [0.9\,\mathrm{k\Omega}, 1.1\,\mathrm{k\Omega}]$ with frequency range 1 Hz to 100 Hz is given by the dashed lines in Fig. 2b. The outer bounds are slightly more pessimistic than the actual range, which can be approximated by classical

[1] this could be shown using the *Sherman-Morrison formula*



(a) RLC-filter schematics



(b) Absolute values of output current: interval AC analysis (dashed lines) and parameter sweeps (solid lines)

Figure 2: RLC filter with component tolerances

parameter sweeps (solid lines). This is due to the wrapping effect mentioned above, and can be reduced by sophisticated bisection methods.

5 NONLINEAR CIRCUITS

The behavior of static nonlinear circuits with uncertain parameters can be described by a system f of nonlinear equations, such that

$$f(\mathbf{x}, \mathbf{p}) = 0, \text{ for all } \mathbf{p} \in [\mathbf{p}]. \qquad (9)$$

Again $\mathbf{x}$ denotes the vector of voltages and currents, and $[\mathbf{p}]$ the interval vector corresponding to the uncertain parameters. If J_f denotes the Jacobian of f then by the *mean value theorem* we get for $\mathbf{x_0}, \mathbf{x}^* \in \mathbb{R}^n$ the following relation:

$$f(\mathbf{x}^*, \mathbf{p}) - f(\mathbf{x_0}, \mathbf{p}) = J_f(\xi, \mathbf{p}) \cdot (\mathbf{x}^* - \mathbf{x_0}), \qquad (10)$$

where ξ is on the line between $\mathbf{x}^*$ and $\mathbf{x_0}$. If $\mathbf{x}^*$ is a zero of f, we get

$$\mathbf{x}^* = \mathbf{x_0} - \Big(J_f(\xi, \mathbf{p})\Big)^{-1} \cdot f(\mathbf{x_0}, \mathbf{p}). \qquad (11)$$

In order to bound the unknown value ξ, it can be replaced by a box $[\mathbf{x}]$, which contains both $\mathbf{x}^*$ and $\mathbf{x_0}$. Hence if we start with a sufficiently large box $[\mathbf{x}]$, the *interval-Newton step*

$$[\mathbf{x}] \leftarrow [\mathbf{x}] \cap \left(\mathbf{x_0} - \Big([J_f]([\mathbf{x}], [\mathbf{p}])\Big)^{-1} \cdot f(\mathbf{x_0}, [\mathbf{p}])\right) \qquad (12)$$

for some real vector $\mathbf{x_0} \in [\mathbf{x}]$, can be used to bound the zeros of f in $[\mathbf{x}]$. For the inversion of J_f a suitable linear contractor can be used. In order to resolve the main source of interval growing, caused by dependent elements of J_f, the attention is focused on matrix inversion. The method of section 4 can be applied to each Newton step, if the nonlinear circuit elements were modeled in a way that the Jacobian has an appropiate structure.

For instance consider a simplified transistor model for the *forward active region* [4], which is based on the bipolar junction transistor model introduced by Ebers and Moll [3]. It expresses the relations between the transistor currents and voltages I_C, I_E, and V_B as:

$$\begin{aligned} I_C &= I_S \cdot e^{(V_B - V_E)/V_t} \\ I_E &= -\tfrac{I_S}{\alpha} \cdot e^{(V_B - V_E)/V_t}, \end{aligned} \tag{13}$$

where the parameter I_S represents the transport saturation current, α denotes the forward current gain of the common base configuration, and V_t designates the thermal voltage. The entries in the Jacobian matrix corresponding to these equations can be written as

$$J_f = p \cdot \begin{pmatrix} \vdots \\ 1 \\ \vdots \\ -\frac{1}{\alpha} \\ \vdots \end{pmatrix} (\cdots 1 \cdots -1 \cdots), \tag{14}$$

introducing a new parameter $p := I_S \cdot \frac{e^{(V_B - V_E)/V_t}}{V_t}$, depending on uncertain values $V_B \in [V_B]$ and $V_E \in [V_E]$ bounded in the previous Newton step and the possibly interval-valued transistor parameter $I_S \in [I_S]$ (global parameters like V_t should be real-valued or of small width). The intermediate value of p can be computed using plain interval arithmetic without running into dependence problems. In every Newton step p may be treated like a linear parameter, and the respective methodology can be applied to it.

Since not all nonlinear effects fit in the scheme described above, classical interval methods still have to be used to get reliable results in these cases. But if the procedure of this section can be applied to a major subeffect, tighter bounds to the solution set will be obtained.

6 CONCLUSIONS

The methods described above are being implemented as an extension to the toolbox *Analog Insydes* [1, 5], an add-on package to the computer algebra system *Mathematica* [10] for modeling, analysis, and design of analog circuits.

The techniques of sections 3 and 4 can be used to obtain meaningful bounds to the results of analog circuit equations with uncertain parameters. As opposed to earlier attempts to use interval arithmetic in this area, which were restricted to the sparse tableau formulation, the dependence between parameters with multiple occurrences is treated accordingly. Finally section 5 shows how this approach is extended to nonlinear equations with a suitable structure.

Despite the limitations in applying interval methods to general equation systems, accordingly tuned interval arithmetic can be used for efficient treatment of the respective circuit classes.

References

[1] Analog Insydes home page. `http://www.analog-insydes.de/`.

[2] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Oxford University Press, Inc., 1986.

[3] J. J. Ebers and J. L. Moll. Large-signal behavior of junction transistors. In *Proc. IRE*, volume 42, pages 1761–1772, Dec. 1954.

[4] R. L. Geiger, E. E. Allen, and N. R. Strader. *VLSI Design Techniques for Analog and Digital Circuits*. Series in electrical engineering. McGraw-Hill, 1990.

[5] T. Halfmann, E. Hennig, M. Thole, and T. Wichmann. *Analog Insydes - Release 2.0 Manual*. Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, Kaiserslautern, Germany.

[6] E. R. Hansen. *Global optimization using interval methods*, volume 165 of *Monographs and textbooks in pure and applied mathematics*. Marcel Dekker, New York, 1992.

[7] R. B. Kearfott. *Rigorous Global Search: Continuous Problems*, volume 13 of *Nonconvex optimization and its applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[8] L. V. Kolev. *Interval Methods for Circuit Analysis*. World Scientific, Singapore, 1993.

[9] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, New York, 2nd edition, 1993.

[10] S. Wolfram. *The Mathematica Book*, volume 4. Wolfram Media/Cambridge University Press, 4th edition, 1999.