Worst-Case Control Mechanism for Approximate Symbolic Analysis
ZDENĚK KOLKA1), VIERA BIOLKOVA1), DALIBOR BIOLEK2)
1)Department of Radioelectronics, Brno University of Technology
2)Department of Electrical Engineering, University of Defence, Brno
CZECH REPUBLIC
kolka@feec.vutbr.cz, biolkova@feec.vutbr.cz, dalibor.biolek@unob.cz

Abstract: The paper deals with a control mechanism for approximate symbolic analysis of the Simplification-Before-Generation class that uses the worst case analysis to incorporate component variations into the ranking mechanism. The traditional approach consists in the evaluation of errors caused by simplification only for the nominal parameters of network components. The proposed procedure uses a fast algorithm for computing the sensitivities needed for the vertex-based worst-case analysis. An advantage of the proposed approach is that the validity of approximated expression is checked over the tolerance interval of component parameters.

Key-Words: Symbolic analysis, Simplification, Error prediction, Worst-case analysis, Linear circuits; CAD tools

1 Introduction
The applicability of exact symbolic analysis in the frequency domain is constrained to relatively small linear circuits, as the size of the resulting expression grows exponentially with the number of nodes and components. If we restrict the range of frequency and network parameters, the majority of symbolic terms can be removed from large expressions without any significant numerical error [1].

Simplification methods can be divided into three classes according to the stage of analysis at which the simplification is performed: Simplification Before Generation (SBG), Simplification During Generation (SDG), and Simplification After Generation (SAG) [2]. The SAG methods are algorithmically simple but very expensive in terms of computation and storage. Methods of the SDG type are rather complex and may have problems with the interpretability of resulting expressions [3]. The SBG methods directly modify the network model. Thus the simplification is inherently more appropriate and intuitive from the point of view of the expression interpretability [3].

Several SBG methods have been proposed so far. They operate with network matrices [2], [4] or graphs [5], [6], [7]. The simplification procedure is essentially the same for all the methods. First, all the prospective simplifications (circuit element removal, matrix or graph modification, etc.) are ranked numerically according to the error their application would cause. One or more operations with the lowest error are actually performed and the numerical solution is updated. The procedure is repeated until the maximum error is reached. All computations of errors should be done numerically, as neither the full nor the simplified functions are available in the symbolic form during the simplification process.

In all cases, an adequate error mechanism is needed to control which term or parameter can be deleted without exceeding some prescribed maximum magnitude and phase errors [1]. Theoretically, the maximum error should be guaranteed on some continuous frequency interval and for network parameters having any values from their tolerance ranges.

The traditional approach consists in the evaluation of errors on a set of frequency samples and for nominal parameters of network components [2], [4]. For a low number of frequency samples there is a danger of exceeding the maximum error between the samples. A method in [6] uses interval analysis techniques to detect such situations. From the point of view of network parameters, the ranking criterion should ideally be based on the Worst-Case analysis. Applicable methods include the interval analysis and the sensitivity-based vertex analysis [8]. A simple application of the interval analysis [9] overestimates the resulting uncertainty due to the intractable interval expansion caused by dependence among interval operands. A tight-interval analysis for linear circuits was proposed, for example, in [10] or [11]. However, these methods were designed to estimate a perturbed solution of the linear system $\mathbf{Ax} = \mathbf{b}$ and cannot be used directly for ranking, which uses the ratio of two perturbed solutions (see Section 2).

Under the assumption of monotonicity the vertex analysis provides that the worst case parameter sets are located at the vertices of parameter space. The method is very fast if the right vertex is identified.
using sensitivities [8].

The method presented in the paper uses the vertex analysis to estimate the worst-case error during a symbolic simplification of the SBG type. It has been developed as an extension of the method in [7] with the aim of achieving the lowest computational cost.

Section 2 of the paper describes the procedure, and Section 3 gives a numerical example.

# 2 Worst-Case Error Estimation

## 2.1 Error of Simplification

The simplification procedure is essentially the same for all methods of the SBG type [2]. It is based on ranking all the prospective operations (parameter or edge removal, node contraction, etc.) according to the error their application would cause. One or more operations with the lowest error are actually performed and the procedure is repeated until the maximum error is reached.

The full symbolic formula provides valid results for any combination of the frequency and parameters of circuit elements, for which the circuit model is valid. The symbolic simplification causes an additional error and restricts the range of validity. Fig. 1 illustrates the procedure where an appropriate selection of control frequencies leads to a first-order approximation. In fact, the approximate symbolic analysis is an interactive process where the designer selects the control frequencies to tune the resulting expression with the help of graphical feedback.

Let $p = [p_1, ..., p_M]$ denote the parameters of network components with associated tolerances

$$p \in D = \left\{ [p_1, ..., p_M] \mid p_i \leq p \leq \bar{p}_i, \ i = 1...M \right\}$$

(1)

where $p_i$ and $\bar{p}_i$ are the lower and the upper bounds, respectively. The symbol $p^0_i$ denotes the nominal value.

Let $F_S(f, p)$ and $F_S(f, \bar{p})$ be the full and the simplified complex network functions, respectively. It is convenient to define the error function as

$$E(f, p) = F_S(f, p) / F_S(f, \bar{p}), \ F_S \neq 0, \ (2)$$

The complex value of (2) represents the magnitude and phase errors in the Bode diagram, i.e.

$$e_{\text{MDB}}(f, p) = 20 \log |E(f, p)|, \ (3)$$

$$e_\phi(f, p) = \arg(E(f, p)). \ (4)$$

Usually, (3) and (4) are evaluated on discrete frequencies and for nominal values of network parameters, i.e. on samples $(f_i, p^0)$. It is expected that the simplified expression will be valid on a “sufficiently” large neighborhood of nominal values.

Let us consider the situation on one control frequency after several iterations of the simplification algorithm. Both errors (3) and (4) will be nonzero due to ongoing simplification, Fig. 2.

The vertex method is based on determining the worst-case combination of bounds of network parameters to achieve the maximum positive or negative deviation of the function being analyzed [8]. Under the assumption of monotonicity the combination leading to maximizing (3) is

$$\text{sgn}\left( \frac{\partial e_{\text{MDB}}(f, p)}{\partial p_i} \right) = \begin{cases} 1 : p_i^\uparrow = \bar{p}_i \\ 0 : p_i^\uparrow = p_i^0 \\ -1 : p_i^\downarrow = p_i \end{cases}, \ (5)$$

and the extreme value of (3) is then simply

$$e_{\text{MDB}} = e_{\text{MDB}}(f, p^\uparrow_{\text{MDB}}), \ (6)$$

where $p^\uparrow_{\text{MDB}}$ is the worst-case combination determined using (5). The combination $p^\downarrow_{\text{MDB}}$ can be determined similarly.

The computation of (5) is based on the knowledge of the sensitivities determined at $p^0$. If a parameter $p$ influences only one network element, it is possible to express any network function $F$ in the bilinear form [12]

$$F = \frac{a p + b}{c p + \bar{d}}, \ (7)$$

where $a, b, c,$ and $d$ are complex coefficients, which depend on the frequency and other parameters. Then
the sensitivities of the magnitude and of the phase of $F$ to parameter $p$ can be expressed as

$$\frac{\partial F|_{ab}}{\partial p} = 20 \log e \text{Re} \left\{ \frac{\partial F}{\partial p} \right\}_1 = 20 \log e \text{Re} \left\{ \frac{a}{ap+b} - \frac{c}{cp+d} \right\},$$

$$\frac{\partial \arg(F)}{\partial p} = \text{Im} \left\{ \frac{\partial F}{\partial p} \right\}_1 = \text{Im} \left\{ \frac{a}{ap+b} - \frac{c}{cp+d} \right\}.$$ (9)

Finally, the sensitivities are

$$\frac{\partial e_{\text{MaB}}}{\partial p} = \frac{\partial F|_{ab}}{\partial p},$$

$$\frac{\partial e_p}{\partial p} = \frac{\partial \arg(F)}{\partial p} - \frac{\partial \arg(F)}{\partial p}.$$ (10)

In order to rank the effect of each simplification step, which changes the value of $p$, a weighted sum of the worst amplitude and the phase errors computed for the worst control frequency is used.

2.2 Effective Algorithm for Bilinear Forms
The circuit being analyzed is represented by a set of linear equations obtained by using, for example, the Modified Nodal Analysis [12]. Without independent sources, we obtain

$$\mathbf{H} \mathbf{x} = 0$$ (12)

where $\mathbf{H}$ is the network matrix, and $\mathbf{x}$ is the vector of unknown voltages and currents.

A parameter $p$ of a single network element can appear in $\mathbf{H}$ in the well-known fill-in patterns [12]. The simplification procedure tries to eliminate the parameter by setting $p \to 0$ or $p \to \infty$. For example, in the case of conductance the element is replaced either by open or by short circuit.

In all cases, the control algorithm has to compute how the change of $p$ affects the numerical value of the network function for each control frequency using the bilinear form (7).

Any network function $F$ can be obtained as a ratio of two algebraic cofactors of $\mathbf{H}$

$$F = \frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)} = \left( \det(\mathbf{H}_1) - p^0 \Delta_1 \right) + p \Delta_1.$$ (13)

Matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are derived from $\mathbf{H}$ by means of adding and deleting some rows and columns [12]. For the parameter $p$ appearing in the four-position pattern in $\mathbf{H}_1$, $\Delta_1 = \Delta_{1(1,1)} + \Delta_{1(1,\infty)} + \Delta_{1(\infty,1)} + \Delta_{1(\infty,\infty)}$ represents the algebraic cofactors of $\mathbf{H}_1$; similarly, $\Delta_2$ represents the cofactors of $\mathbf{H}_2$. If $p$ appears in one or two positions, $\Delta_1$ and $\Delta_2$ are expressed just with one or two cofactors.

The algebraic cofactors can be obtained by a simple matrix inversion using

$$C_1 = \det(\mathbf{H}_1) \mathbf{H}_1^{-1} \mathbf{H}_2^{-1}, \quad C_2 = \det(\mathbf{H}_2) \mathbf{H}_1^{-1} \mathbf{H}_2^{-1},$$ (14)

where each element of $C_1$ and $C_2$ is the respective algebraic cofactor of $\mathbf{H}_1$ and $\mathbf{H}_2$.

At this point, formula (13) allows predicting the new value of $F$ for $p \to 0$ or $p \to \infty$ for the nominal values of parameters.

In order to estimate worst-case errors (3) and (4) it is necessary to have bilinear form (7) for all remaining parameters that have not been eliminated in the previous steps. To do so we need updated cofactor matrices (14) for $p \to 0$ or $p \to \infty$. Theoretically, it would require two new matrix inversions.

In the case when only a few elements of the matrix were changed, the Sherman-Morrison formula can be used to update the original inverse matrix [13]. Let

$$\tilde{\mathbf{H}}_1 = \mathbf{H}_1 + q \mathbf{u} \mathbf{v}^T$$ (15)

be the updated matrix $\mathbf{H}_1$. Vectors $\mathbf{u}$ and $\mathbf{v}$ define the coordinates of the updated matrix elements. In the case of the four-position fill-in pattern of $q$ the vectors are simply $\mathbf{u} = [0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0]$, $\mathbf{v} = [0,...,0,1,0,0,0,0,0,0,0,0,0,...,0]$. The new inverse is then

$$\tilde{\mathbf{H}}_1^{-1} = \mathbf{H}_1^{-1} - \frac{q}{1+q^2 \mathbf{u} \mathbf{v}^T} \mathbf{H}_1^{-1} \mathbf{u} \mathbf{v}^T \mathbf{H}_1^{-1}.$$ (16)

For $p \to 0$, a compensating term $q = p^0$ should be added to the matrix. Then

$$\tilde{\mathbf{H}}_{00}^{-1} = \mathbf{H}_1^{-1} + \frac{p^0 (\mathbf{H}_{1(1,1)} - \mathbf{H}_{1(1,\infty)} \mathbf{H}_{1(\infty,1)} - \mathbf{H}_{1(\infty,\infty)})}{1 - p^0 (\mathbf{H}_{1(1,1)} + \mathbf{H}_{1(1,\infty)} - \mathbf{H}_{1(\infty,1)} - \mathbf{H}_{1(\infty,\infty)})},$$ (17)

and for $p \to \infty$

$$\tilde{\mathbf{H}}_{\infty}^{-1} = \mathbf{H}_1^{-1} - \frac{(\mathbf{H}_{1(1,1)} - \mathbf{H}_{1(1,\infty)} \mathbf{H}_{1(\infty,1)} - \mathbf{H}_{1(\infty,\infty)})}{\mathbf{H}_{1(1,1)} + \mathbf{H}_{1(1,\infty)} - \mathbf{H}_{1(\infty,1)} - \mathbf{H}_{1(\infty,\infty)}},$$ (18)

where $\mathbf{H}_{1(i,j)}$ is the $i$-th column of $\mathbf{H}_1$, $\mathbf{H}_{1(i,*)}$ is the $i$-th row, and $\mathbf{H}_{1(i,j)}$ is the coefficient $(i,j)$ of the matrix. The same formulae hold for $\mathbf{H}_2$. Similar formulae can be derived if the parameter $q$ appears in the two- or one-position pattern.

Having established the new inverse matrices it is possible to compute the new cofactor matrices

$$\tilde{\mathbf{C}}_{00} = (\det(\mathbf{H}_1) - p^0 \Delta_1) \mathbf{H}_{1(0,0)}^{-1} \mathbf{H}_{1(1,0)}^{-1}, \quad \tilde{\mathbf{C}}_{\infty} = \Delta_1 \mathbf{H}_{1(\infty,0)}^{-1} \mathbf{H}_{1(1,0)}^{-1}.$$ (19)
and use them to obtain bilinear form (7) for all the remaining parameters in order to compute the worst-case values (6).

The procedure of ranking the removal operation of one parameter is summarized in Fig. 3. The reference numerical solution for the full function should be computed before the simplification process starts.

01: compute cofactors (14);
02: compute bilinear form (13) for selected \( p \);
03: compute nominal errors (3), (4) for \( p \to 0 \)
04: if (errors not acceptable) {
    compute nominal errors (3), (4) for \( p \to \infty \);
05: if (errors not acceptable) {
    06: mark \( p \) as “necessary”; exit;
07: }
08: }
09: update cofactors (19) for \( p \to 0 \) or \( p \to \infty \);
10: compute worst-case combinations (5) and error (6) and use them for ranking;

Fig. 3 Computation of worst-case ranking.

### 2.3 Numerical cost

Generally, the dimension of circuit matrix \( H \) and the number of network parameters decrease during the simplification. The estimated numerical complexities are assumed for the original circuit size.

Line 01: The computation of one cofactor matrix (14) for one control frequency requires \( O((n-1)^3) \) operations. The determinant is a byproduct of the matrix inversion. Cofactor matrices (14) are common to all tested parameters and should be recomputed after each iteration of the main cycle.

Line 09: The update (19) of one cofactor matrix requires \( O((n-1)^3) \) long operations. This line has to be recomputed for each parameter to be eliminated.

Line 10: Worst-case errors are computed by solving the original and the simplified systems, i.e. they require \( O(n^3) \) operations if the parameter removal is acceptable for nominal values.

### 3 Example

Let us consider a bridge with error amplifier with a differential-pair frontend, Fig. 4.

Ignoring the amplifier input impedance, the voltage transfer \( V_{out}/V_{in} \) depends both on the differential gain \( A_D \) and on the common-mode gain \( A_{CM} \):

\[
K = \frac{(R_1 R_4 - R_2 R_3) A_D + 0.5 (R_3 R_4 + 2 R_1 R_3 + R_2 R_4) A_{CM}}{(R_1 + R_2)(R_3 + R_4)}.
\]

In the symmetrical case \( R_1 = R_3, R_2 = R_4 \), there are two large terms, \( R_1 R_4 \) and \( R_2 R_3 \), in (20) that cancel each other. Thus the numerical value of (20) will be insensitive to the changes of \( A_D \) the simplification procedure would make, i.e. parameters that are significant for the differential voltage gain may be eliminated from the model. The problem has been known as the “large-term cancellation” [5].

Parameters of passive elements are shown in Fig. 4. All the transistors are N-types from the AMI 13780 technology with an (equivalent) aspect ratio of 64/2 for M1 and M2, and 32/2 for M3 and M4. The supply voltage was 3.3V, and the reference current \( I_{ref} = 100\mu A \). The tolerances of all parameters were set to 10%.

Using one reference frequency \( f_1 = 1Hz \) with a magnitude tolerance of \( \Delta M_1 = 1dB \), the procedure gives a correct answer for the DC gain

\[
K = \frac{R_1 g_{m1} g_{m2} (R_1 R_4 - R_2 R_3)}{(g_{m1} + g_{m2}) (R_1 + R_2)(R_3 + R_4)},
\]

where \( g_{m1} \) and \( g_{m2} \) are the transconductances of M1 and M2, respectively. Fig. 5 shows the result obtained from the experimental software. The transfer function is shown in the plain format. The full symbolic form has 288 terms in the numerator and 1334 terms in the denominator.

Fig. 5 Simplified result for DC gain with worst-case frequency characteristic.

---

**Fig. 4 Balanced bridge with error amplifier.**

**Fig. 5 Balanced bridge with error amplifier.**
5 Conclusions

The proposed procedure uses a fast algorithm to obtain sensitivities for the vertex analysis. Taking into account component parameter variations yields better results in comparison with a simple computation for nominal parameters only.

Acknowledgements

This work has been supported by the Czech Science Foundation under grant No P102/10/1665, by the project CZ.1.07/2.3.00/20.0007 WICOMT of the operational program Education for competitiveness, and Project for the development of K217 Department, UD Brno – Modern electrical elements and systems.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No 230126.

References:


