

Algorithm for estimating error of symbolic simplification

Zdenek Kolka^{1,*}, Viera Biolkova¹, Josef Dobes², and Martin Horak¹

¹Brno University of Technology, Department of Radio Electronics, Brno, Czech Republic

²Czech Technical University in Prague, Department of Radioelectronics, Prague, Czech Republic

Abstract. The paper deals with an improved algorithm for estimating errors during approximate symbolic analysis. A linear system can be solved symbolically. However, the size of the resulting formula grows exponentially with the matrix size. The approximate symbolic analysis omits insignificant terms of the exact formula to decrease its size, which, on the other hand, limits the validity of the approximate result. The proposed algorithm estimates, in a computationally feasible way, the approximation error over a region of system parameters. This makes it possible to maintain the validity of the results even if the tolerances of the system parameters are defined. The method is based on the first-order approximation of error functions. The algorithm is demonstrated using the SNAP symbolic analyzer, which has been developed by the authors.

1 Introduction

A broad class of technical problems can be solved successfully by means of approximation by a linear dynamical system. Let us take as examples all the linear analog signal processing in electrical engineering, mechanical systems within the scope of Hooke's law, hydraulic systems and others. Such a dynamical system can be subjected to the Fourier or Laplace transforms, which lead to a system of linear equations that allow a symbolic solution. In contrast to a purely numerical solution, the symbolic formula provides a better insight into the system behavior and allows using the obtained information also for synthesis [1].

The symbolic analysis was originally developed in electrical engineering. Therefore the proposed algorithm will be explained on the example of electrical circuit analysis. However, the results can be easily generalized to other physical domains based on analogies [2].

A system of linear equations can be easily solved symbolically by means of graph-based or algebraic algorithms [3]. The main problem is that the complexity of the resulting symbolic formula, measured by the number of components, grows exponentially with the system size. Therefore the symbolic analysis is computationally unfeasible for most practical problems in terms of time and storage. In addition, a huge analytical formula cannot be interpreted nor used for speeding-up the calculations because its complexity exceeds the complexity of a purely numerical solution.

* Corresponding author: kolka@feec.vutbr.cz

A feasible way to get smaller symbolic expressions is to reduce accuracy to achieve simplicity, i.e. to use the approximate symbolic analysis [1]. The method mimics designers' work, when they simplify the system model by omitting negligible phenomena (e.g. by omitting stray-capacitance currents on low frequencies, negligible mass in mechatronic systems, etc.). The simplification is performed by omitting some symbolic terms of negligible numerical influence from the initially exact expression. For example, a parallel combination of two resistors $R_1 \ll R_2$ can be simplified to $R_1 R_2 / (R_1 + R_2) \approx R_1$ after omitting R_1 in the denominator.

The symbolic analysis starts with a formulation of the system of linear equations, followed by the algebraic or graph-based solution and the symbolic formula processing. The simplification can be applied in each stage of the analysis and the respective methods are called Simplification Before/During/After Generation (SBG/SDG/SAG) [3].

The simplification of the system of (complex) linear equations (SBG) is the most powerful one, as the method deals with a relatively small system of equations [4]. In addition, the equation simplification admits a physical interpretation (e.g. small current/force/pressure omission, etc.), while the other methods, SDG and SAG, rather resemble a mathematical approximation, which is harder to interpret in terms of the original physical system [5].

It is obvious that symbolic simplification has an impact on the validity of results. Whereas the full formula is valid for all combinations of system parameters, the simplified one introduces an error and its validity is limited to a certain region of parameters.

Considering the steady-state harmonic analysis, a maximum simplification error should be guaranteed on a continuous interval of frequencies and on a continuous region of network parameters. Unfortunately, no finite method is known to ensure this requirement. The usual solution is based on zero-order approximation, where the error is checked only for nominal values of network parameters and it is "believed" that the expression will be valid in a sufficiently large vicinity of nominal values [3]. An application of the interval analysis to estimating the influence of component tolerances was proposed in [6]. However, the interval analysis tends to overestimate the resulting uncertainty and is difficult to perform for large systems.

This paper proposes using the first-order approximation of error functions to estimate the simplification error for network parameters within their tolerance ranges. Section 2 provides theoretical and implementation details, and Section 3 presents two example problems solved using the SNAP symbolic simulator, which has been developed by the authors [7].

2 Error estimation procedure

2.1 Linear system and its parameters

Let us consider a steady state harmonic analysis of a lumped linear dynamical system using the frequency-domain operational calculus. The analysis leads to a system of complex linear equations

$$\mathbf{H}(j\omega, \mathbf{p})\mathbf{x} = \mathbf{b}, \quad (1)$$

where $\mathbf{H} \in \mathbb{C}^{N \times N}$, $\mathbf{b} \in \mathbb{C}^N$ is the system matrix and the vector of sources, $\mathbf{x} \in \mathbb{C}^N$ are the unknown quantities, and $\mathbf{p} \in \mathbb{R}^K$ is the vector of system parameters (parameters of its components). The solution of (1) is usually characterized by a transfer function, which is a ratio of two quantities

$$F(j\omega, \mathbf{p}) = \frac{\mathbf{x}_i}{\mathbf{x}_j}. \quad (2)$$

This function will be the subject of simplification.

Let us consider a closed region D of network parameters

$$D: \left| \mathbf{p}_i - \mathbf{p}_i^{(nom)} \right| \leq \Delta \mathbf{p}_i, \quad \forall i = 1 \dots K, \quad (3)$$

where $\mathbf{p}^{(nom)}$ are the nominal values of network parameters with symmetric tolerances $\Delta \mathbf{p}$.

No finite method exists that would assure the validity of the approximation of (2) over a closed frequency interval $\langle f_1, f_2 \rangle$ and the region D . Practical experience shows that it is sufficient to check the error on several discrete frequencies from $\langle f_1, f_2 \rangle$. In fact, a suitable choice of frequency samples determines the character of simplification results. As will be shown in Section 3, the user can focus on a particular band, and a simple frequency-domain plot provides feedback on the circuit behavior between samples or outside $\langle f_1, f_2 \rangle$.

2.2 Simplification procedure

Let us consider, for simplicity, that the system (1) will be simplified by means of omitting some parameters from \mathbf{p} that have an insignificant influence on the transfer function (2). Let $F_R(j\omega, \mathbf{p})$ be the original exact transfer function and $F_A(j\omega, \mathbf{p})$ be the approximate function at some stage of simplification. To simplify the notation, the product $j2\pi$ of the expression $j\omega = j2\pi f$ is absorbed into the functions F_R and F_A .

First, each possible modification of (1) is assigned a weight based on the distance between F_R and F_A in a suitable metric that this modification would cause. One or more modifications with the lowest weight are actually performed and the list of modifications and their weights will be updated. The process terminates, when no further simplification is possible due to excessive error. The process is depicted in Fig. 1 in a pseudocode.

```

compute numerical reference  $F_R$ ;
while not finished {
    weight all possible simplification operations;
    if no simplification is possible, then exit;
    perform operation(s) with the lowest weight;
}
    
```

Fig. 1. Main cycle of simplification procedure.

The key point is to establish an appropriate metric in the transfer function space, which is computationally feasible and sufficiently accurate with respect to (3).

2.3 Metric for transfer function

The basic tool for evaluating frequency analysis results is the Bode diagram, i.e. two separate plots for magnitude in decibels and phase in (usually) degrees. The chosen metric is based on differences between F_R and F_A in both diagrams. Let us define the magnitude- and phase-error functions depending on frequency and the parameters \mathbf{p}

$$e_M(f, \mathbf{p}) = 20 \log |F_A(f, \mathbf{p})| - 20 \log |F_R(f, \mathbf{p})| = 20 \log |F_A(f, \mathbf{p}) / F_R(f, \mathbf{p})|, \quad (4)$$

$$e_\varphi(f, \mathbf{p}) = \arg(F_A(f, \mathbf{p})) - \arg(F_R(f, \mathbf{p})) = \arg(F_A(f, \mathbf{p}) / F_R(f, \mathbf{p})). \quad (5)$$

Note that for practical problems, neither F_A nor F_R are available in symbolic form in the early stages of simplification due to computational complexity, and both (4) and (5) should be evaluated numerically.

Let ΔM_i and $\Delta \varphi_i$ be respectively the magnitude and the phase error limits specified for each frequency sample f_i , $i = 1, \dots, M$. Then, the cumulative weight can be defined as

$$w = \max_{i=1..M} \left\{ \frac{\max_{\mathbf{p} \in D} |e_M(f_i, \mathbf{p})|}{\Delta M_i} + \frac{\max_{\mathbf{p} \in D} |e_\varphi(f_i, \mathbf{p})|}{\Delta \varphi_i} \right\}. \quad (6)$$

The value of (6) grows during the simplification as it accumulates all the previous steps and the process is terminated when it reaches the limit value.

The choice of frequency samples f_i is left to the user. Based on the display of the Bode diagram, one can chose the frequencies manually. Finding the maximum weight for any $\mathbf{p} \in D$ cannot be done manually due to the high number of parameters. The prosed solution is based on the sensitivity of the error functions (4) and (5) to all parameters in \mathbf{p} .

It is well known that any transfer function (2) of the lumped conservative system (1) can be expressed with respect to a chosen parameter p_i as bilinear function [8]

$$F(f, p_i) = \frac{a(f, \tilde{\mathbf{p}}) p_i + b(f, \tilde{\mathbf{p}})}{c(f, \tilde{\mathbf{p}}) p_i + d(f, \tilde{\mathbf{p}})}, \quad (7)$$

where a, b, c, d are complex coefficients depending on frequency and the remaining parameters $\tilde{\mathbf{p}} = \mathbf{p} \setminus \{p_i\}$. The coefficients can be obtained numerically by means of cofactors of the matrix \mathbf{H} in (1) [8].

The bilinear expansion (7) can be done both for F_R and F_A and we get

$$e_M(f, p_i) = 20 \log |F_A(f, \mathbf{p}) / F_R(f, \mathbf{p})| = 20 \log \left| \frac{a_A p_i + b_A}{c_A p_i + d_A} \frac{a_R p_i + b_R}{c_R p_i + d_R} \right|, \quad (8)$$

and a similar expression for e_φ .

With respect to the following equalities

$$\frac{d(20 \log |z|)}{dp} = 20 \log e \operatorname{Re} \left\{ \frac{dz}{z} \frac{1}{z} \right\}, \quad \frac{d(\arg(z))}{dp} = \operatorname{Im} \left\{ \frac{dz}{z} \frac{1}{z} \right\} \quad (9)$$

for a complex variable z and a real parameter p , we finally obtain

$$\frac{de_M}{dp_i} = 20 \log e \operatorname{Re} \left\{ \frac{a_A}{a_A p_i + b_A} - \frac{c_A}{c_A p_i + d_A} - \frac{a_R}{a_R p_i + b_R} + \frac{c_R}{c_R p_i + d_R} \right\}, \quad (10)$$

$$\frac{de_\varphi}{dp_i} = \operatorname{Im} \left\{ \frac{a_A}{a_A p_i + b_A} - \frac{c_A}{c_A p_i + d_A} - \frac{a_R}{a_R p_i + b_R} + \frac{c_R}{c_R p_i + d_R} \right\}. \quad (11)$$

Thus it is possible to compute the analytical derivatives of error functions based on the knowledge of numerical algebraic cofactors of the system matrix \mathbf{H} .

Under the assumption of monotonicity of e_M and e_φ over the region D , the maximum in (6) can be estimated using the well-known corner (or Worst-Case) analysis [9]. The maximum positive and negative deviations of the error function from the nominal value occur for a certain combination of extreme deviations of the parameters \mathbf{p} depending on the signs of the derivatives (10) and (11). For e_M the worst-case combinations will be

$$\mathbf{P}_M^\uparrow = \left\{ p_1^{(nom)} + \operatorname{sgn}\left(\frac{de_M}{dp_1}\right)\Delta p_1, \dots, p_K^{(nom)} + \operatorname{sgn}\left(\frac{de_M}{dp_K}\right)\Delta p_K \right\}, \quad (12)$$

$$\mathbf{P}_M^\downarrow = \left\{ p_1^{(nom)} - \operatorname{sgn}\left(\frac{de_M}{dp_1}\right)\Delta p_1, \dots, p_K^{(nom)} - \operatorname{sgn}\left(\frac{de_M}{dp_K}\right)\Delta p_K \right\}, \quad (13)$$

and similarly for e_φ . The maximum error in (6) can be estimated as “the worst” of both corners (12) and (13)

$$\max_{\mathbf{p} \in D} |e_M(f_i, \mathbf{p})| \approx \max\left(|e_M(f_i, \mathbf{P}_M^\uparrow)|, |e_M(f_i, \mathbf{P}_M^\downarrow)|\right), \quad (14)$$

$$\max_{\mathbf{p} \in D} |e_\varphi(f_i, \mathbf{p})| \approx \max\left(|e_\varphi(f_i, \mathbf{P}_\varphi^\uparrow)|, |e_\varphi(f_i, \mathbf{P}_\varphi^\downarrow)|\right). \quad (15)$$

The monotonicity of e_M and e_φ can be verified by comparing the signs of derivatives for nominal values and for the corners (12) and (13). If they do not agree, there is a possibility of the existence of local extrema within D . However, there is no finite method to find the extrema.

2.4 Computational complexity

The algebraic cofactors for computing (7) can be obtained by a simple matrix inversion

$$\mathbf{C}(j\omega, \mathbf{p}) = \det(\mathbf{H})(\mathbf{H}^{-1})^T, \quad (16)$$

where \mathbf{C} is the matrix of algebraic cofactors of \mathbf{H} . Because the determinant is a byproduct of the matrix inversion, the computational complexity of (16) can be estimated as $O(N^3)$ long operations and the complexity of (6) as $O(MN^3)$.

3 Example analyses

The proposed method will be illustrated by two examples. Both circuits were analyzed using our symbolic analyzer SNAP [7], which has been continuously developed since 1995.

3.1 Large-term cancellation

The first circuit illustrates the potential danger of calculating errors only for nominal values of the circuit parameters. The input voltage is applied to a resistive bridge, which is balanced for nominal parameters ($R_1 = R_2 = R_3 = R_4$). Its diagonal voltage is amplified by a transconductance amplifier U1 with the transconductance $g_m = 10$ mS. The output node sums the output currents of the amplifier and the feedforward contribution through R_F . The circuit is simple enough for the complete symbolic formula for input-to-output voltage transfer to be, after some arrangements, written as

$$F_R = \frac{g_m R_F (R_2 R_3 - R_1 R_4) + (R_1 + R_2)(R_3 + R_4)}{(R_1 + R_2)(R_3 + R_4)} \frac{R_o}{R_F + R_o}. \quad (17)$$

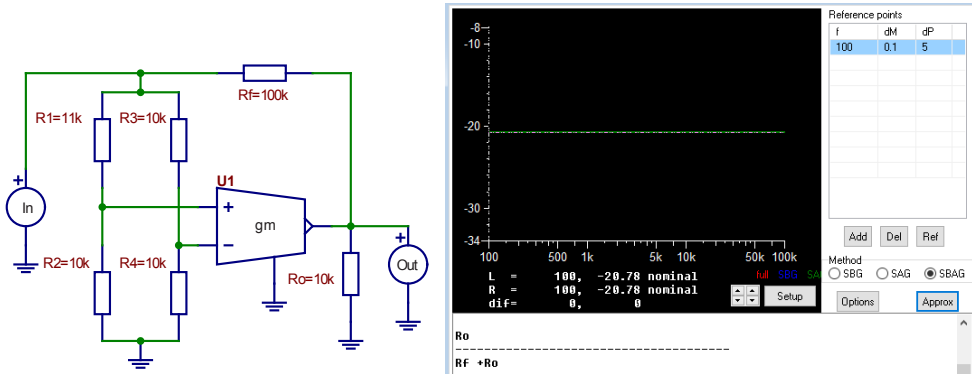


Fig. 2. Bridge amplifier with feedforward path and simplified transfer function for nominal parameters for $\Delta_M = 0.1\text{dB}$. (“In” and “Out” pseudocomponents denote the input and output ports.)

It is obvious that for nominal parameters the input voltage of the transconductance amplifier is zero. Thus the complete removal of the amplifier by setting $g_m = 0$ will not change the output voltage.

The transfer function does not depend on frequency, i.e. the frequency of the reference point can be arbitrary and the phase will always be zero. If the maximum allowed magnitude error in (6) is $\Delta_M = 0.1\text{dB}$, the simplification process in Fig. 1 will in the case of considering only nominal parameters yield

$$F_A = \frac{R_o}{R_F + R_o}, \tag{18}$$

which is, in fact, the transfer of a resistive divider formed by R_F and R_o . However, in the case of unbalanced bridge the contribution of the signal path through the amplifier is substantial and formula (18) no longer describes the behavior of the real circuit.

Let us consider the testing of $g_m = 0$ (i.e. what happens in the case of removing U1). For nominal parameters the magnitude error will be $e_M = 0\text{dB}$ but not its sensitivity. For example, for $p_i = R_1$ the bilinear coefficients (7) of the simplified transfer function will be

$$a_A = 0, \quad b_A = R_o, \quad c_A = 0, \quad d_A = R_o + R_F, \tag{19}$$

and the coefficients of the full function will be

$$\begin{aligned} a_R &= R_o(R_3 + R_4 - g_m R_F R_4), & b_R &= R_o[g_m R_F R_2 R_3 + R_2(R_3 + R_4)], \\ c_R &= (R_3 + R_4)(R_F + R_o), & d_R &= R_2(R_3 + R_4)(R_F + R_o). \end{aligned} \tag{20}$$

Note that (19) and (20) can be shown here in symbolic form because of the circuit simplicity. In the case of practical-size problems the coefficients are available only numerically, i.e. by using cofactors.

Using (10), (19) and (20) we can compute the sensitivity of e_M to R_1 as $de_M/dR_1 = 0.217 \text{ dB}/\Omega$. Even a one-Ohm increase in R_1 causes an increase in the transfer that is greater than the previously allowed $\Delta_M = 0.1\text{dB}$. For a 10% tolerance of R_1, R_2, R_3 , and R_4 no simplification is possible and (17) is the correct formula.

3.2 Error function estimation

The second circuit in Fig. 3 has been chosen to demonstrate the effect of control frequency selection and the corner-based estimation of phase and magnitude errors. It represents a classical active RC low-pass configuration with $f_c = 1.4$ kHz. U1 represents a fully symbolic model of a single-pole operating amplifier including the input capacitance (C_{in}), output resistance (R_o), and transfer function

$$K_{OpAmp} = \frac{A_o}{1 + s \frac{A_o}{2\pi B_1}}, \quad (21)$$

where A_o is the DC gain and B_1 is the unity-gain bandwidth.

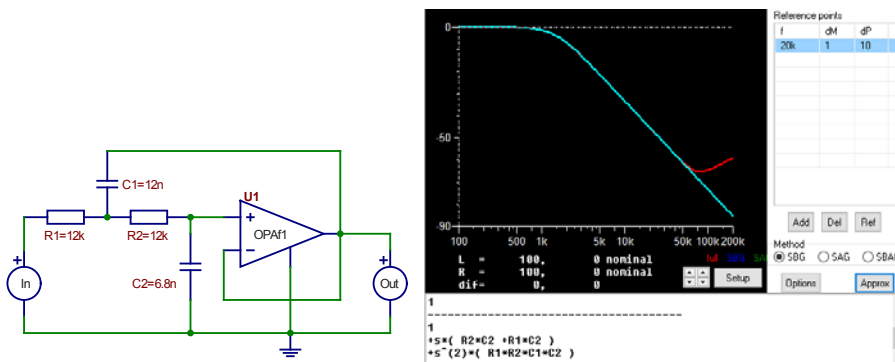


Fig. 3. Active low-pass filter and its approximate analysis with the reference point at 20 kHz ($\Delta_M = 1$ dB, $\Delta_\phi = 10^\circ$). OpAmp model: $A_o = 2 \cdot 10^5$, $B_1 = 3$ MHz, $C_{in} = 3$ pF, $R_o = 200 \Omega$.

The approximate analysis results depend on the choice of the control frequencies, i.e. where the error is checked. If the frequency was set, for example, to 100 Hz, the result would be “1”, as the low-frequency transfer of the filter does not depend on any component value. The procedure would remove C_1 and C_2 , shorten R_1 and R_2 , and replace the operating amplifier with the ideal one.

Fig. 3 shows another choice of the frequency at 20 kHz with $\Delta_M = 1$ dB and $\Delta_\phi = 10^\circ$. The resulting transfer function, shown as the cyan curve in the SNAP output, can be obtained with the ideal opamp in the filter. The simplification procedure replaced the amplifier with the ideal amplifier as its parasitic properties were not significant at 20 kHz. The red curve in Fig. 3 is the reference solution with the full amplifier model, which shows the effect of nonzero output resistance above 100 kHz. The approximate symbolic formula is the well-known transfer function of the low-pass Sallen-Key filter.

The reference point at 20 kHz is relatively close to the parasitic transmission zero at 80 kHz and therefore the variance of opamp parameters should be taken into account. The following tolerances were used: $\Delta_{A0} = \Delta_{B1} = \Delta_{R0} = \Delta_{Cin} = 30 \%$, $\Delta_{R1} = \Delta_{R2} = 1\%$, $\Delta_{C1} = \Delta_{C2} = 5 \%$. Fig. 4 shows the last three steps of the simplification. The blue box shows the worst-case error estimation using the corner analysis. The red circle represents error functions for the nominal parameters. The image of parameter tolerances in the e_M - e_ϕ plane cannot be easily obtained and the Monte-Carlo analysis with 1000 samples and uniform distribution was used instead to estimate statistically the real influence of tolerances.

The influence of eliminating $C_{in} \rightarrow 0$ and $A_o \rightarrow \infty$ (i.e. the opamp is modelled as an integrator) is negligible and the nominal point is almost in the origin in the chosen scale. As the simplification proceeds, the nominal red point moves away from the origin. It can be

clearly seen that even the modest component tolerances cannot be ignored because the maximum errors are more than double the nominal ones. The phase error is more significant here and the influence of simplification on magnitude is negligible.

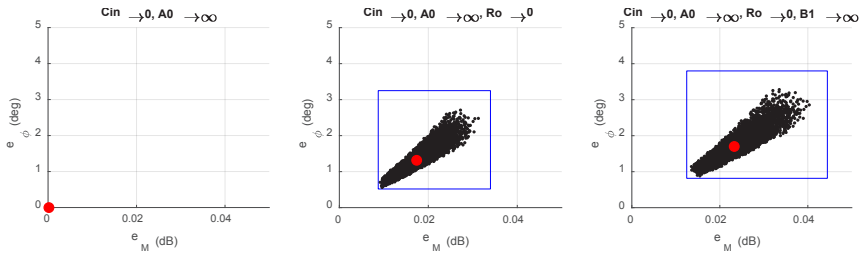


Fig. 4. Comparison of worst-case corners with the Monte-Carlo simulation of component tolerances.

4 Conclusions

The worst-case corner analysis provides a simple and computationally inexpensive means of estimating the effect of element tolerances on the validity of the simplified result. The first example also demonstrates the potential danger of using nominal values only. On the other hand, the required condition of monotonicity cannot be easily verified.

This work has been supported by the Czech Science Foundation under grant agreement No. 20-26849S.

References

1. M. Fakhfakh, E. Tlelo-Cuautle, F.V. Fernández, *Design of Analog Circuits through Symbolic Analysis* (Bentham Science, Soest, NL, 2012)
2. F.E. Cellier, M. Otter, H. Elmquist, *Proc. of ICBGM'95* (Las Vegas, 1995, 49-55)
3. E. Hennig, *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits* (Ph.D. Dissertation, Univ. of Kaiserslautern, 2000)
4. Z. Kolka, M. Vlk, M. Horák, *Radioengineering*, **20**(1), 252-256 (2011)
5. W. Daems, G. Gielen, W. Sansen, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, **21**(4), 395-407 (2002)
6. F.V. Fernández, A. Rodríguez-Vázquez, J.D. Martín, and J.L. Huertas, *Int. J Analog Int. Circ. and Signal Proc.*, **3**(1) 43–85 (1993)
7. Z. Kolka, D. Biolek, J. Kalous, V. Biolková, *Proc. of SM2ACD 2010* (Tunis (Tunisia), 2010, 120-123)
8. W.K. Chen, *Active Network Analysis* (World Scientific Publishing, 1991)
9. M.W. Tian, C.J.R. Shi, *IEEE Trans. on Circuits and Systems I*, **47**(8), 1138-1145 (2000)