

Approximate Symbolic Analysis with Emphasis on Frequency Filters

ZDENEK KOLKA¹, VIERA BIOLKOVA¹, DALIBOR BIOLEK²

¹Department of Radio Electronics, ²Department of Microelectronics

Brno University of Technology

Purkyňova 118, 612 00 Brno

CZECH REPUBLIC

kolka@feec.vutbr.cz, biolkova@feec.vutbr.cz, dalibor.biolek@unob.cz

Abstract: The paper deals with two error control strategies for approximate symbolic analysis in the frequency domain: the response-based approach and the eigenvalue-based approach. The response-based mechanism, which evaluates errors caused by simplification by means of comparing magnitude and phase responses with a reference solution on a set of fixed frequencies, is suitable for broadband circuits with a flat response. In case of frequency-selective circuits (i.e. filters) the simple method may not provide expected results. The eigenvalue-based method uses shifts of poles and zeros in the complex plane to evaluate the effect of circuit model simplifications. The position of poles and zeros in the complex plane provides better characterization of the frequency response of a frequency filter. An effective implementation of both methods is presented in the paper.

Key-Words: Symbolic analysis, simplification, error prediction, eigenvalues, linear circuits.

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]. Negligible symbolic terms are identified numerically, based on the known parameters of circuit components.

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 SBG methods simplifying the circuit equations or graphs are the most effective ones, as they work with a relatively small number of circuit equations. Since SBG techniques directly operate on the network model, the simplification is inherently more appropriate and intuitive from the point of view of the expression interpretability [3], [9].

Several SBG methods have been proposed so far. They operate with network matrices [4], [5] or graphs [6], [7], [8]. 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, Fig. 1.

```
compute reference numerical solution;
while  $\varepsilon_A < \varepsilon_{\max}$  {
    generate all possible simplification operations;
    compute the error of each operation;
    perform operation(s) with the lowest error;
    update numerical solution and  $\varepsilon_A$ ;
}
undo last operation;
```

Fig. 1. Main cycle of SBG.

In all cases, an adequate error mechanism is needed to control which term or parameter can be deleted without exceeding some prescribed errors [1], [5], [9]. The majority of published solutions are focused on broadband circuits, i.e. amplifiers in particular. In this case, it is sufficient if the control algorithm evaluates amplitude and phase errors on a fixed set of frequencies [10]. For a low number of frequency samples there is a danger of exceeding the maximum error between the samples. A method in [7] uses interval analysis techniques to detect if the error is exceeded in some intermediate frequency. A method presented in [10] uses small-change sensitivities to estimate the worst-case error during symbolic simplification of the SBG type.

Fig. 2 shows how a suitable choice of one or two control frequencies leads to zero- or first-order

approximation of transfer or immittance function.

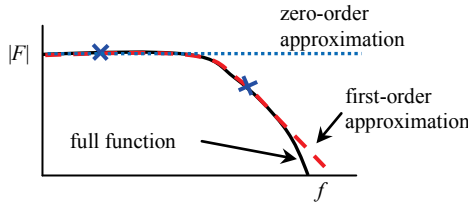


Fig. 2. Effect of selection of control frequencies on resulting approximation.

In case of frequency-selective circuits (i.e. filters), checking the magnitude and phase response on a relatively low number of test frequencies may not be sufficient. Such circuits are better characterized by the position of poles and zeros in the complex plane. For transistor-level circuits there are a lot of parasitic roots with negligible influence on the response in the frequency interval of interest. Therefore, the analysis of practical circuits requires performing the simplification in two main steps:

1. Decrease of the transfer-function order using classical amplitude- and phase-control procedure, which removes the majority of negligible poles and zeros.
2. Simplification of SBG type with eigenvalue-based control procedure, which monitors the position of selected poles and zeros.

Section 2 of the paper deals with an effective implementation of the response based method and the eigenvalue based method. Section 3 presents an illustrative example.

2 Implementation of control procedures

2.1 Response-based procedure

The simplification introduces an error whose maximum value is checked at several selected frequencies with specified magnitude and phase tolerances ΔM_{ω_i} and $\Delta \varphi_{\omega_i}$.

Assuming m test frequencies ω_i the response-based error criterion is

$$\varepsilon_{A,r} = \max_{i=1..m} \left\{ \frac{20 \log |E(\omega_i)|}{\Delta M_{\omega_i}} + \frac{\arg(E(\omega_i))}{\Delta \varphi_{\omega_i}} \right\}, \quad (1)$$

where $E(\omega) = F_A(\omega)/F_R(\omega)$, and $F_A(\omega)$ and $F_R(\omega)$ are simplified and reference network functions, respectively. The error of approximation ε_{Ar} is, in fact, a weighted sum of magnitude and phase errors in the Bode diagram. The choice of appropriate test frequencies is left to the user (see Fig. 2).

In the simplest case, the simplification procedure tries to eliminate parameter p of a circuit element by setting $p \rightarrow 0$ or $p \rightarrow \infty$. For example, in the case of

conductance the element is replaced either by open or by short circuit [8].

The control algorithm has to compute how the change of the parameter p affects the numerical value of the network function for each control frequency. This can be done easily using the bilinear expansion of network function, which can be obtained easily following the classical procedure of matrix-based circuit analysis

$$F = \frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)} = \frac{(\det(\mathbf{H}_1) - p_{nom}\Delta_1) + p\Delta_1}{(\det(\mathbf{H}_2) - p_{nom}\Delta_2) + p\Delta_2}. \quad (2)$$

Matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are derived from network matrix $\mathbf{H}$ by means of adding and deleting some rows and columns [11]. Symbol $\Delta_1 = \Delta_{H1i:i} + \Delta_{H1j:j} - \Delta_{H1i:j} - \Delta_{H1j:i}$ represents the algebraic cofactors of $\mathbf{H}_1$, Δ_2 represents the cofactors of $\mathbf{H}_2$, and p_{nom} is the nominal value of the parameter.

If the determinants and cofactors are known, it is easy to compute how F changes for any value of p , including infinity, by using (2). The algebraic cofactors can be obtained by a simple matrix inversion

$$\mathbf{H}_{1,\Delta} = \det(\mathbf{H}_1)(\mathbf{H}_1^{-1})^T, \quad \mathbf{H}_{2,\Delta} = \det(\mathbf{H}_2)(\mathbf{H}_2^{-1})^T, \quad (3)$$

where each element of $\mathbf{H}_{1,\Delta}$ and $\mathbf{H}_{2,\Delta}$ is the respective algebraic cofactor. The computation of one cofactor matrix requires approximately $O(n^3)$ operations, where n is the actual matrix size, i.e. the numerical cost corresponds to the cost of the AC analysis.

2.2 Eigenvalue-based procedure

Transfer or immittance function of any lumped linear time-invariant circuit is determined up to a real multiplicative constant by the position of poles and zeros. The pole-zero analysis leads to the generalized eigenvalue problem [12]

$$\mathbf{v}^H (\mathbf{A} - \lambda \mathbf{B}) = \mathbf{0} \quad \text{or} \quad (\mathbf{A} - \lambda \mathbf{B})\mathbf{u} = \mathbf{0}, \quad (4a,b)$$

where λ is the eigenvalue (i.e. pole or zero), $\mathbf{v}$ and $\mathbf{u}$ are left and right eigenvectors, respectively. The upper index H means the Hermitian transpose. The procedure for obtaining matrix pencil $(\mathbf{A}, \mathbf{B})$ from the circuit matrix is similar both for poles and zeros and can be found, for example, in [12].

It is obvious that any modification of the circuit model during the simplification process causes shifts of all eigenvalues. A simple computing of the eigenvalue spectra after each circuit modification by means of the QZ or an equivalent algorithm is not feasible due to computational cost and due to the problem of matching the original and the modified

eigenvalues.

A more appropriate solution is provided by the Jacobi-Davison method [13], which allows a selective correction of a chosen eigenpair $(\mathbf{u}, \lambda)$ for perturbed matrix pencil. This gives the possibility to track eigenvalue shifts with moderate computational effort.

The method approximates the exact solution $(\mathbf{u}_p, \lambda_p)$ of the perturbed eigenproblem $(\mathbf{A}_p, \mathbf{B}_p)$ by iteration process

$$(\mathbf{u}^{(i+1)}, \lambda^{(i+1)}) = f(\mathbf{u}^{(i)}, \lambda^{(i)}), \mathbf{u}^{(0)} = \mathbf{u}_0, \lambda^{(0)} = \lambda_0 \quad (5)$$

where $(\mathbf{u}_0, \lambda_0)$ is the solution of the original eigenproblem $(\mathbf{A}_0, \mathbf{B}_0)$.

Let us suppose $(\mathbf{u}^{(i)}, \lambda^{(i)})$ lies in the vicinity of expected solution $(\mathbf{u}_p, \lambda_p)$. Then the eigenvalue update will be [13]

$$\lambda^{(i+1)} = \frac{\mathbf{w}^H \mathbf{A}_p \mathbf{u}^{(i)}}{\mathbf{w}^H \mathbf{B}_p \mathbf{u}^{(i)}}, \quad (6)$$

where $\mathbf{w}$ is a vector whose choice will be specified later.

The residuum of (4b) is

$$\mathbf{r} = \mathbf{A}_p \mathbf{u}^{(i)} - \lambda^{(i+1)} \mathbf{B}_p \mathbf{u}^{(i)}. \quad (7)$$

The eigenvector update is

$$\mathbf{u}^{(i+1)} = (\mathbf{u}^{(i)} + \mathbf{z}) / \|\mathbf{u}^{(i)} + \mathbf{z}\|, \quad (8)$$

where $\mathbf{z}$ is obtained as a solution of linear system

$$\begin{bmatrix} \mathbf{A}_p - \lambda^{(i+1)} \mathbf{B}_p & \tilde{\mathbf{w}} \\ \tilde{\mathbf{u}}^H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ x \end{bmatrix} = \begin{bmatrix} -\mathbf{r} \\ 0 \end{bmatrix}, \quad (9)$$

where x is an auxiliary variable.

The choice of vectors $\mathbf{w}$, $\tilde{\mathbf{w}}$, and $\tilde{\mathbf{u}}$ influences convergence of the iteration process. There are several possibilities [13]. For example, the choice of $\tilde{\mathbf{u}} = \mathbf{w} = \mathbf{v}^{(0)}$, $\tilde{\mathbf{w}} = \mathbf{u}^{(0)}$ leads to a fast convergence.

The iteration process is stopped when the norm of the residuum vector is lower than a chosen margin

$$\|\mathbf{r}\| \leq r_{stop}. \quad (10)$$

Then the measure of an individual eigenvalue shift is defined as

$$\varepsilon_{A,e} = \frac{|\lambda^{(0)} - \lambda^{(N)}|}{|\lambda^{(0)}|}, \quad \lambda^{(0)} \neq 0, \quad (11)$$

where N is the number of iterations after which the iterating process was stopped.

3 Example Analysis

The procedure from Section 2 has been implemented experimentally in Matlab. The test analysis was performed for the low-pass filter from Fig. 3 with behavioral model of the opamp.

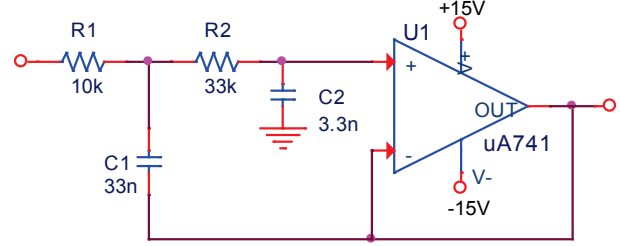


Fig. 3. Sallen-Key low-pass filter.

In the pole-zero diagram, there is a pair of conjugated poles $(-1.97 \cdot 10^3 \pm j4.88 \cdot 10^3)$, which determines the desired low-pass frequency response of the biquad. In addition, there is a pair of low-frequency zeros $(141.63 \cdot 10^3 \pm j250.07 \cdot 10^3)$, which causes the -70 dB trough at 60 kHz.

Locking the control procedure on the zeros yields approximated transfer function

$$K_{app} = \frac{\omega_1 A_0 + s^3 (R_O R_2 C_2 C_1)}{s^2 (\omega_1 A_0 R_1 R_2 C_1 C_2)}. \quad (12)$$

where ω_1 and A_0 characterize the dominant pole and DC gain of the opamp. It can be easily seen there are, in fact, three zeros distributed on a circle with radius $\sqrt[3]{\omega_1 A_0 / (R_O R_2 C_2 C_1)}$.

Fig. 4 shows comparison of transfer function (12) with the original circuit. It can be easily seen that (12) is valid only for obtaining zeros. The poles determining the low-frequency behavior of the filter were shifted to the origin by the simplification procedure.

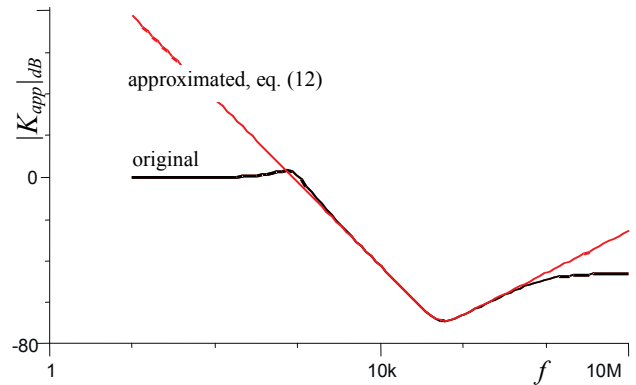


Fig. 4. Comparison of original and approximated transfer functions for filter from Fig. 2

4 Conclusions

The paper deals with two control procedures for approximate symbolic analysis of the Simplification-Before-Generation class. The amplitude-based method is suitable for circuits with a simple dominant frequency response, while the eigenvalue-based method is suitable in case of complex responses. Both methods can be combined in the analysis flow for transistor-based models, whose complexity requires the application of fast response-based method at first. After reducing the initial model, the eigenvalue control procedure can be used.

The approximate symbolic analysis is an interactive process, whose result strongly depends on the choice of parameters of the control procedure.

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:

- [1] W. Daems et al., Evaluation of Error-Control Strategies for the Linear Symbolic Analysis of Analog Integrated Circuits. *IEEE Trans Circ. Syst.-I*, vol. 46, no. 5, 1999, pp. 594-606.
- [2] V. Fernández, A. Rodríguez-Vázquez, L. Huertas, G. Gielen, *Symbolic analysis techniques*. New York: IEEE press, 1998.
- [3] W. Daems, G. Gielen, W. Sansen, Circuit Simplification for the Symbolic Analysis of Analog Integrated Circuits. *IEEE Trans Comp Aided Design*, vol. 21, no. 4, 2002, pp. 395-407.
- [4] R. Sommer et al., Equation-Based Symbolic Approximation by Matrix Reduction with Qualitative Error Prediction. *Alta-Frequenza Rivista di Elettronica*, vol.5, no.6, Dec. 1993, pp.29-37.
- [5] J. Hsu, C. Sechen, DC small signal symbolic analysis of large analog integrated circuits. *IEEE Trans. on Circ Syst-I*, vol. 41, no. 12, 1994, pp. 817-828.
- [6] Q. Yu, C. Sechen, A Unified Approach to the Approximate Symbolic Analysis of Large Analog Integrated Circuits. *IEEE Trans Circ Syst-I*, vol. 43, no. 8, 1996, pp. 656-69.
- [7] O. Guerra, J. Rodriguez-Garcia, E. Roca, F. Fernández, A. Rodríguez-Vázquez, A simplification before and during generation methodology for symbolic large-circuit analysis. In *Proc. of IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS'98)*, 1998, pp. 81-84.
- [8] Z. Kolka, D. Biolek, V. Biolková, M. Horák, Implementation of Topological Circuit Reduction. In *Proc. of the 11th IEEE Asia Pacific Conference on Circuits and Systems*, Malaysia, 2010, pp. 951-954.
- [9] E. Tlelo-Cuautle et al., Behavioral Modeling of Mixed-Mode Integrated Circuits. *Advances in Analog Circuits*, ed. E. Tlelo-Cuautle, Rijeka (Croatia): InTech, 2011, pp. 85-108.
- [10] Z. Kolka, D. Biolek, V. Biolková, J. Dobeš, Tolerance-based Control Mechanism for Approximate Symbolic Analysis. In *Proceedings of the IEEE Region 8 AFRICON 2011 conference*, Livingstone, Zambia: IEEE, 2011. pp. 1-4.
- [11] W. K. Chen, *Active Network Analysis*, World Scientific Publishing, UK, 1991.
- [12] J. Vlach, *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, New York, 1993.
- [13] G.L.G. Sleijpen, A.G.L. Booten, D.R. Fokkema, H.A. van der Vorst, Jacobi-Davidson Type Methods for Generalized Eigenproblems and Polynomial Eigenproblems. *BIT Numerical Mathematics*, vol. 36, no. 3, pp. 595-633.