Accurate Time-Domain Semisymbolic Analysis

Zdenek Kolka, Dalibor Biolek, Viera Biolkova
Faculty of Electrical Engineering and Communication
Brno University of Technology
Brno, Czech Republic
kolka@feec.vutbr.cz

Abstract— **The paper deals with a method for accurate semisymbolic time-domain analysis of highly idealized linear lumped circuits. Pulse and step responses can be computed by means of the partial fraction decomposition. The procedure relies on an accurate computation of poles of the transfer function. The well known problem of the *QR* and *QZ* algorithms is their poor accuracy in the case of multiple roots. Moreover, the partial fraction decomposition itself is an ill-posed problem for closely-spaced clusters of roots. The method presented in this paper is based on an improved reduction procedure for transforming the generalized eigenproblem into a standard one in combination with an algorithm for computing the Jordan canonical form of inexact matrices [1].**

Keywords-linear circuits; inverse Laplace transform; eigenvalues; pulse and step responses

I. INTRODUCTION

Pulse and step responses of lumped linear systems can be expressed semisymbolically, i.e. as a formula which, in addition to the numerical constants, also contains the symbol of time t. For a transfer function in rational form the inverse Laplace transform can be performed using the well-known partial fraction decomposition. The procedure relies on an accurate computation of poles of the transfer function and on reliable identification of pole multiplicity.

The pole-zero analysis is very sensitive to the rounding of input data as well as to numerical errors of floating-point operations. The problem appears naturally during the analysis of large systems and also during the analysis of relatively small but highly idealized circuits that often lead to multiple eigenvalues. As shown in [2], computational errors can have severe consequences for the semisymbolic analysis of time-domain responses.

One of the most difficult problems is the computation of multiple roots. In fact, the question whether there exists a multiple eigenvalue is an ill-posed problem [3]. Using any numerical precision we always get a cluster of eigenvalues. The method of secondary root polishing [2] applies a heuristic procedure to identify multiple roots from closely-spaced clusters. But the result still depends on the accuracy of computed roots.

This work has been supported by the Czech Science Foundation under grant No. GA 102/08/0784 and by the Czech Ministry of Education under research project OC09016, which is a part of the COST Action IC0803 RFCSET. 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.

There is an accuracy limit to computing multiple roots [4]: to calculate an m-fold root to the precision of k correct digits, the accuracy of the coefficients and the machine precision must be at least $m \cdot k$ digits. This property naturally leads to the use of multiprecision arithmetic to maintain the required accuracy, but at the cost of highly increased computational time [5].

The first step of the time-domain semisymbolic analysis is the pole-zero analysis, which leads to the generalized eigenvalue problem. We will use our enhanced procedure [6] for reducing the matrix pencil of the generalized problem to a standard one. The reduction uses the Singular Value Decomposition for explicit rank estimation with the aim of avoiding spurious roots.

Subsequently, the resulting standard eigenvalue problem is solved using a new algorithm of Z. Zeng for computing the Jordan canonical form of inexact matrices [1]. Unlike classical methods, which treat multiple roots as a collection of single ones, the algorithm [1] estimates the multiplicity structure first and then finds the respective eigenvalues using a well-conditioned iteration scheme. The result is an unprecedented accuracy that breaks the attainable accuracy barrier.

Section 2 of the paper describes the algorithm and Section 3 gives some numerical examples.

II. TIME-DOMAIN SEMISYMBOLIC ANALYSIS

A. Basic concept

Let $F(s)$ be a network function of the circuit being analyzed. The pulse response $g(t)$ and the step response $h(t)$ can be obtained using the inverse Laplace transform

$$g(t) = \pounds^{-1}\{F(s)\},\ \ h(t) = \pounds^{-1}\{F(s)/\mathrm{s}\}. \qquad (1)$$

Let $Y(s)$ be a function given either as $Y(s) = F(s)$ or as $Y(s) = F(s)/s$ depending on the response to be computed. For lumped circuits we obtain

$$Y(s) = \frac{P(s)}{Q(s)} = K\frac{\widetilde{P}(s)}{\prod_{i=1}^{r}(s-p_i)^{m_i}} + d_0 + d_1 s + \ldots, \qquad (2)$$

where p_i is generally a complex pole with multiplicity m_i, r is the number of distinct poles, and K is a real multiplying coefficient. The fraction on the right-hand side of (2) is proper,

978-1-4244-6815-7/10/$26.00 ©2010 IEEE

i.e. the degree of the numerator is lower than the degree of the denominator,

$$\sum_{i=1}^{r} m_i < \deg(\widetilde{P}(s)) \ . \tag{3}$$

The numerator $\widetilde{P}(s)$ can be obtained as a remainder after the long division of $P(s)$ by $Q(s)$. The real quotients d_i of the long division directly lead to the Dirac pulses in the output response.

The fractional part of (2) can be decomposed into a sum of partial fractions. Due to the linearity of the Laplace transform the time-domain response will be a sum of inverse transforms of individual partial fractions.

For a pole p_i we obtain m_i partial fractions in the form

$$\frac{c_1}{s-p_i}+\frac{c_2}{(s-p_i)^2}+\ldots+\frac{c_{m_i}}{(s-p_i)^{m_i}} \ . \tag{4}$$

The coefficients c_i are generally complex. An accurate algorithm for the expansion, which does not require the evaluation of polynomial coefficients, can be found in [7].

The inverse Laplace transform of one partial fraction from (4) for a real pole is

$$\frac{c_j}{(s-p_i)^j} \doteq \frac{c_j}{(j-1)!}t^{j-1}e^{p_i t}, \ t \geq 0 \ . \tag{5}$$

For a complex conjugated pair $p_i = \sigma_i + j\omega_i$, $\overline{p}_i = \sigma_i - j\omega_i$ fractions (4) appear in conjugated pairs. The inverse transform of one pair leads to

$$\frac{c_j}{(s-p_i)^j}+\frac{\overline{c}_j}{(s-\overline{p}_i)^j} \doteq \frac{2|c_j|}{(j-1)!}t^{j-1}e^{\sigma_i t}\cos\left(\omega_i t+\arg(c_j)\right), \tag{6}$$

The numerical computation of multiple roots is an ill-posed problem. Using any numerical precision we always get a cluster of eigenvalues. A typical situation is illustrated in Fig. 1 for a zero eigenvalue with multiplicity 5 computed using the *QZ* method [8].

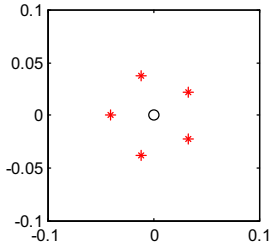

Figure 1. The typical effect of finite precision on multiple eigenvalue. (circle – 5-fold eigenvalue, asterisk – numerically determined cluster)

The partial fraction decomposition is numerically unstable for such closely-spaced clusters. Let us assume that the eigenvalue algorithm found a cluster of single or conjugated poles instead of a multiple one. Then, the Heaviside method can be used to find the coefficients of expansion

$$\frac{\widetilde{P}(s)}{Q(s)} = \sum_{i=1}^{n} \frac{\widetilde{P}(p_i)}{Q'(p_i)(s-p_i)} \ , \tag{7}$$

where p_i are distinct poles (roots of Q), and Q' is the derivative of Q with respect to the Laplace operator s. According to the Gauss-Lucas theorem [9] the roots of the polynomial derivative lie within the convex hull of the original roots of Q, i.e. within the cluster around the exact multiple eigenvalue. Thus the value of $Q'(p_i)$ is given by numerical errors of the eigenvalue algorithm, which is essentially an arbitrary number.

To overcome the problem, various polishing methods have been proposed. In principle, they heuristically identify closely-spaced clusters as one multiple eigenvalue. Nevertheless, the result always depends on the (poor) accuracy of computed roots.

The next chapter describes briefly an application of a novel algorithm [1], which estimates the multiplicity structure first and then finds the respective eigenvalues using a well-conditioned iteration scheme. As a result, the partial fraction decomposition is well-conditioned too.

B. Accurate analysis of multiple roots

The pole-zero analysis leads to the generalized eigenvalue problem. As the available algorithm [1] is only for standard eigenvalue problems, the first step begins with transforming the matrices.

Let (**A**, **B**) be a matrix pencil whose eigenvalues are zeros or poles of a network function. The procedure for obtaining (**A**, **B**) from the circuit matrix can be found in [10]. Let us write

$$\mathbf{C}(s) = \mathbf{A} - s\mathbf{B} \tag{8}$$

Our goal is to transform in two steps the matrix pencil into a matrix **Q** which has (approximately) the same eigenvalues. The first step converts the matrix **C** to form (9) using a procedure similar to the Gaussian elimination, and the second step reduces the matrix $\mathbf{C}^{(\mathrm{I})}$ to form (10) by the pivotal condensation. **I** is the identity matrix.

$$\mathbf{C}^{(\mathrm{I})}(s) = \begin{pmatrix} \mathbf{A}_{11} - s\mathbf{I} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} , \tag{9}$$

$$\mathbf{C}^{(\mathrm{II})}(s) = \mathbf{Q} - s\mathbf{I} \ . \tag{10}$$

Although the eigenvalues of **C** can be well-defined, performing the conversion in finite precision often produces rounding errors, which lead to additional spurious eigenvalues of very big absolute values. To eliminate the effect, we compute beforehand the rank of **B** by the Singular Value Decomposition, which is known to be the most reliable algorithm for this task [11]. Details of the reduction procedure can be found in our paper [6].

The reduction algorithm takes the linear matrix pencil and transforms it into a single matrix **Q**, whose eigenvalues approximate the finite eigenvalues of the original problem. This matrix is then passed to the algorithm *ApproxiJordanForm* for computing the Jordan canonical form of inexact matrices, which is a part of the *ApaTools* package [12]. First, the procedure estimates the eigenvalue multiplicity structure (or equivalently finds the Jordan blocks) and then computes the eigenvalues, using a well-conditioned iteration process [1].

III. NUMERICAL EXAMPLES

The proposed method of semisymbolic time-domain analysis was tested in Matlab (v7.8). Figure 2 shows a benchmark circuit - two-stage filter designed using the FIR-BL (BiLinear) approximation [13]. Considering ideal operational amplifiers and the following equalities

$C_{11} = C_{21} = C_{31} = C_{12} = C_{22} = C_{32} = C$,
$R_{11} = R_{21} = R_{31} = R_{41} = R_{21} = R_{22} = R_{32} = R_{42} = R$,
$R_{51} = R_{52} = 4\,R_{21} = 4\,R_{22}$

the transfer function of each stage is

$$K_i(s) = a_i \frac{s^2 + \omega_{zi}^2}{(s+\omega_0)^2},\ i = 1,2 \qquad (11)$$

where

$$a_i = \frac{R_{4i}}{R_{6i}},\ \omega_{zi} = \frac{1}{RC}\sqrt{\frac{R_{6i}}{R}},\ i = 1,2,\ \omega_0 = \frac{1}{2RC}\ . \qquad (12)$$

The transfer function has two conjugate zeros and one quadruple real pole

$$\begin{aligned} z_{1,2} &= \pm j7.98953027764 \times 10^5 \\ z_{3,4} &= \pm j2.54050871091 \times 10^5 \\ p_{1,2,3,4} &= -5.61797752809 \times 10^4 \end{aligned} \qquad (13)$$

Additionally, the matrix pencil of the denominator of the voltage transfer function is defective, i.e. the eigenspace dimension is only one whereas the pole is fourfold.

Table I shows the results of benchmark analysis of poles for different numbers n of sections obtained using the *QZ* algorithm of Matlab and the reduction method with approximate Jordan decomposition (*RedJord*), see Section II. The exact pole value was computed using (12).

In the ideal case, for n sections we obtain a $2n$-fold pole. As the *QZ* algorithm provides only a cluster of poles, the error shown is the maximum relative distance from $-\omega_0$, $\max\|p_i + \omega_0\|$. The *RedJord* algorithm found the pole multiplicity correctly in all cases, providing accuracy comparable to the machine precision (*eps*) of *double* variables. Deviations in accuracy of the *RedJord* algorithm in Table I are in the order of machine *eps*. The *QZ* algorithm lost its precision quickly for higher multiplicities.

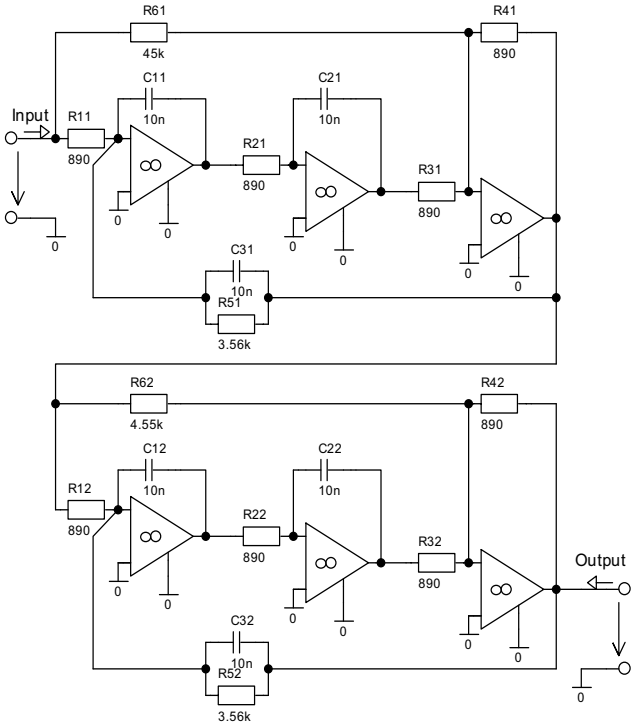

Figure 2. Idealized two-stage active filter.

TABLE I. MAXIMUM RELATIVE ERROR OF COMPUTED POLES

n	**1**	**2**	**3**
QZ	2.6×10^{-16}	1.6×10^{-8}	4.5×10^{-5}
RedJord	3.9×10^{-16}	1.2×10^{-15}	1.8×10^{-15}
n	**4**	**5**	**6**
QZ	4.0×10^{-3}	1.8×10^{-2}	2.1×10^{-2}
RedJord	0.0	3.9×10^{-16}	0.0

Figure 3 shows the ability of *RedJord* to separate nearby multiple roots. The circuit in Fig. 2 was analyzed with different values of capacitors for each section

$C_{11} = C_{21} = C_{31} = C_1$,
$C_{12} = C_{22} = C_{32} = C_2$.

The value of C_1 was firmly set to 10 nF, while the value of C_2 was swept in close vicinity of C_1 from 0.9985 to 1.0015 of the nominal value of 10nF. The poles are then

$$p_{1,2} = -\frac{1}{2RC_1}\ ,\ p_{3.4} = -\frac{1}{2RC_2}\ . \qquad (14)$$

Theoretically, it is only for $C_2 = 10$nF that the circuit has a quadruple pole. For other values it has two double poles. The horizontal axis in Fig. 3 is the exact distance between $p_{1,2}$ and $p_{3,4}$ during the sweep of C_2 computed using (14). The vertical axis shows the position of numerically determined $p_{1,2}$ and $p_{3,4}$ relatively to the theoretical value of $p_{1,2}$ (14). If the distance between poles is smaller than a certain value, which can be

specified by the user, the nearby double poles are identified as one quadruple pole.

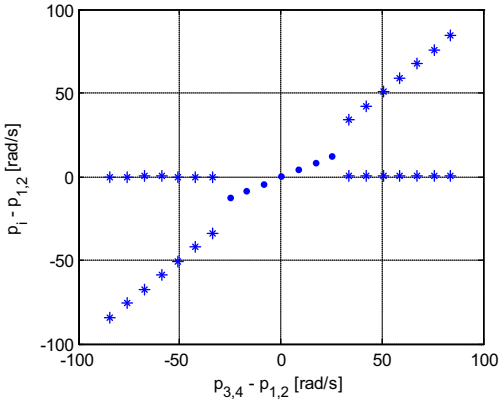

Figure 3. Analysis of two double poles by *RedJord*. (asterisk–double pole, dot–quadruple pole)

The pulse response of the circuit in Fig. 2 in case of two double real poles p_1 and p_2 will be in semisymbolic form

$$g(t) = d_1\delta(t) + c_{1,1}e^{p_1 t} + c_{1,2}te^{p_1 t} + c_{2,1}e^{p_2 t} + c_{2,2}te^{p_2 t}, \qquad (15)$$

where δ(t) is the Dirac pulse and $c_{i,j}$ are the coefficients of partial fraction expansion. Table II shows poles computed for C_1 = 10 nF and C_2 = 9.994985 nF (the first point of Fig. 3) and the coefficients of partial fraction decomposition computed using algorithm [7]. The exact values of $c_{i,j}$ were obtained from analytical form of (15) using Maple. The difference between the exact response and the response computed with *RedJord* was 346, i.e. 0.18% at the peak, while the difference for the *QZ* response was 1.7% apart from the fact that the structure of poles was wrong. The occurrence of a complex pair in results of the *QZ* algorithm leads to oscillating terms in the pulse response (15), i.e. there will be a complex pair of coefficients $c_{1,2}$. All the results were computed with standard IEEE 754 double-precision arithmetic.

TABLE II. COMPUTED POLES AND PFD COEFFICIENTS

RedJord	
poles	*coefficients*
$-5.617977528093498\times10^4$ (double)	$c_{1,1} = -1.4463383375688\times10^{17}$
	$c_{1,2} = 6.10280499938053\times10^{18}$
$-5.626417153818045\times10^4$ (double)	$c_{2,1} = 1.44633833756655\times10^{17}$
	$c_{2,2} = 6.10374996265219\times10^{18}$
***QZ** (Matlab)*	
poles	*coefficients*
$-5.626417153820619\times10^4$ (double)	$c_{1,1} = 1.44633833402522\times10^{17}$
	$c_{1,2} = 6.10374995294316\times10^{18}$
$-5.617977528089886\times10^4$ $\pm j\ 9.427184263548398\times10^{-4}$	$c_{2,1} = -7.23169167013737\times10^{16}$ $\mp j\ 3.23681218990624\times10^{21}$

IV. CONCLUSIONS

The advantage of the method presented if well-defined behavior in the proximity of transfer functions with multiple roots. Numerical experiments have shown the numerical accuracy to be maintained even for defective circuit matrices with high multiplicity roots.

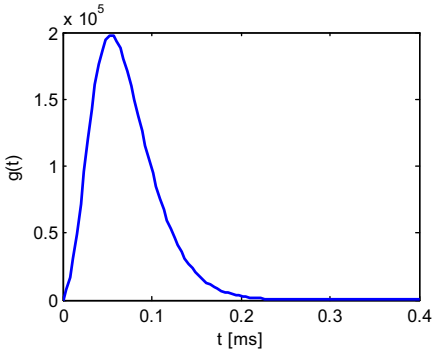

Figure 4. Pulse response of circuit in Fig. 2 for C_1 = 10 nF and C_2 = 9.994985 nF (without the Dirac pulse).

REFERENCES

[1] Z. Zeng, "Numerical Algorithm for Computing the Jordan Canonical Form." In Proc. of Recent Advances in Numerical Methods for Eigenvalue Problems (RANMEP2008), Taiwan, 2008.

[2] D. Biolek, V. Biolkova, "Secondary Root Polishing: Increasing the Accuracy of Semisymbolic Analysis of Electronic Circuits," WSEAS Trans. on Mathematics, 2004, vol. 3, no. 3, pp. 493 - 497.

[3] S.M. Rump, "Computational Error Bounds for Multiple or Nearly Multiple Eigenvalues." Linear Algebra and its Applications, 2001, vol. 324, pp. 209–226.

[4] V.Y. Pan, "Solving polynomial equations: some history and recent progress," SIAM Review, 39 (1997), pp. 187-220.

[5] J. Dobes, J. Michal, "Using the Variable-Length Arithmetic for an Accurate Poles-Zeros Analysis." Radioengineering, 2003, vol. 12, no. 3, p. 1-5.

[6] Z. Kolka, M. Horák, D. Biolek, V. Biolková, "Accurate Semisymbolic Analysis of Circuits with Multiple Roots." In Proc. of the 13th WSEAS int. conference on Circuits, CSCC2009. Greece, 2009. p. 178-181.

[7] D. Westreich, "Partial Fraction Expansion without Derivative Evaluation." IEEE Trans. on CAS, Vol.38, No.6, June 1991, pp.658-660.

[8] C.B. Moler, G.W. Stewart, "An algorithm for generalized matrix eigenvalue problems," SIAM J. Numer. Anal., 10, p.241-256, 1973.

[9] J.L. Díaz-Barrero, J. J. Egozcue, "A generalization of the Gauss-Lucas theorem," Czechoslovak Mathematical Journal, vol. 58, no. 2, June, 2008, p. 481-486.

[10] J. Vlach, Computer Methods for Circuit Analysis and Design. Van Nostrand Reynhold, New York, 1993.

[11] W.H. Press, S.A. Teukolski, W. T. Vetterling, B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Second Edition. Cambridge university press, USA, 2002.

[12] Z. Zeng, "ApaTools: A Software Toolbox for Approximate Polynomial Algebra." Software for Algebraic Geometry (ed. M. E. Stillman, N. Takayama, J. Verschelde), Springer New York, 2008, pp. 149-167.

[13] D. Biolek, V. Biolková, "FIR-BL Analog Filters." In: Proceedings of the 42nd Midwest Symposium on Circuits and Systems MWSCAS'99, Las Cruces, New Mexico, USA 1999, pp. 1021-1023.