Secondary Root Polishing: Increasing the Accuracy of Semisymbolic Analysis of Electronic Circuits

DALIBOR BIOLEK VIERA BIOLKOVA*
Dept. of Microelectronics *Radioelectronics
Brno University of Technology
Purkynova 118, 612 00 Brno
CZECH REPUBLIC
dalibor.biolek@vabo.cz http://www.vabo.cz/stranky/biolek

Abstract – The so-called secondary root polishing techniques are described. They are used in the SNAP program to eliminate numerical inaccuracies during the zero/pole computation of the circuit functions of linear and linearized circuits.

Key-Words: - root polishing, multiple root, eigenvalue problem, SNAP

1 Introduction

The operation of some programs for the analysis of linear and linearized circuits are based on the computation of zeros and poles of circuit functions. The SNAP program for the symbolic, semisymbolic, and numerical analyses belongs to this class [1]. The final analysis accuracy then strongly depends on the accuracy of algorithms for finding zeros and poles. The question of accuracy of these algorithms is fundamental especially in some cases, partly in the analysis of large circuits of the order of about 100 and more, partly in circuits having a considerable spread of time constants and natural frequencies, or in circuits with multiple poles [2].

In principle, zeros/poles are usually computed by two different methods: either from the circuit matrix by the solution of generalized eigenvalue problem, or from the numerator/denominator polynomials by finding their roots. Generally, the first method is more accurate and - if possible – we always prefer it.

Nevertheless, in some cases of analysis, the zero/pole computation represents a numerically sophisticated problem, and even the very accurate algorithms can generate inadmissible errors. To decrease these errors, special techniques, called *root polishing*, are often used [3]. There are methods of iterative root improvement, which sometimes make it possible to decrease the error to a level which is given by the roundoff processor error. Let us call all these familiar methods *primary root polishing techniques*.

Let us assume that instead of a correct couple of poles [-1,-1] we get after the primary polishing [-1,-0.999999999999]. This tiny error (10^{-10} %) causes an incorrect identification of two different poles instead of one twofold. For instance, then the equation of the step response will be of quite a different structure than in the correct case.

The above example is a typical illustration of an error which can be eliminated by the so-called *secondary root polishing techniques*.

2 Techniques of secondary polishing

2.1 Polishing by zeroing the negligible components

Parameters

½Min½/½Max½, ½Re/Im½, ½Im/Re½, ½Re½, ½Im½

control the zero and pole polishing using the method of zeroing the negligible components. This polishing removes the numerical inaccuracy, as given in the following examples:

set of roots

before polishing	**after polishing**
-3.213458697e-34+j2.398128934e12	+j2.398128934e12
-3.213458697e-34 -j2.398128934e12	-j2.398128934e12
-8.872653489e9+j1.9900029898	-8.872653489e9
2.339947872e-9	0

Description of the above techniques will be given for poles of the circuit functions. For zeros, the procedure is equivalent.

In the first step, the maximum absolute value *MAX* of all poles is computed.

Criterion ½**Min**½/½**Max**½: Zoering of poles with "too small" absolute values.

All the poles whose absolute values are smaller than

*MAX**½**Min**½/½**Max**½

are set to zero.

From the physical point of view, the absolute values of roots represent the natural frequencies of first- or second-order subcircuits, which model the frequency response of analyzed circuit. That is why the *MAX* gives the maximum of all natural frequencies, and ½**Min**½/½**Max**½ gives the relationship between the minimum and maximum frequency. The default value

½**Min**½/½**Max**½=100n

implies that all natural frequencies which are less than the maximum one by at least 7 decades are not considered in the frequency response.

In some cases, when the analyzed circuit has a larger spread of natural frequencies, ½**Min**½/½**Max**½could be decreased. For example,

½**Min**½/½**Max**½=1e-m

defines the interval of *m* decades between the minimum and the maximum natural frequencies.

In the SNAP program, this parameter can be modified from 0 to 10m. The value 0 means that polishing is disabled.

Criteria ½**Re/Im**½, ½**Im/Re**½, ½**Re**½, ½**Im**½: Zeroing of negligible real or imaginary parts of poles.

Generally, every pole is a complex number *a*+*jb*.

if	**then**
abs(*a*) < abs(*b*)*½**Re/Im**½	*a* = 0
abs(*b*) < abs(*a*)*½**Im/Re**½	*b* = 0
abs(*a*) < ½**Re**½	*a* = 0
abs(*b*) < ½**Im**½	*b* = 0

From the physical point of view, the real parts of roots represent the reciprocal values of time constants *t* of first- or second-order subcircuits which model the transient responses of analyzed circuit. The imaginary parts represent angular frequencies $\omega=2\pi/T$ of oscillations of the given subcircuits in their responses to quick changes in the exciting signals.

From this point of view, the inequalities given above represent the following relations:

$$wt = 2p\,\frac{t}{T} > \frac{1}{|\mathrm{Re/Im}|},$$

$$wt = 2p\,\frac{t}{T} < |\mathrm{Im/Re}|,$$

$$t > \frac{1}{|\mathrm{Re}|},$$

$$w < |\mathrm{Im}|.$$

Criterion ½**Re/Im**½ (½**Im/Re**½) eliminates too large/small relationships *t*/*T*. To set the implicit values 10 of both parameters, we replace all poles whose relationship *t*/*T* is more than 15915 by their real parts, and all poles whose relationship *t*/*T* is less than 15915 by their imaginary parts.

In some cases, e.g. in the analysis of resistive circuits, while modeling the influence of parasitic reactances, parameter ½**Re/Im**½ should be decreased. On the contrary, analyzing losses of idealized reactance circuits may sometimes require decreasing of ½**Im/Re**½.

Criterion ½**Re**½ eliminates too large time constants *t*, criterion ½**Im**½ too small oscillation frequencies *w*. Setting the implicit values 10m for both parameters, we replace all poles whose time constants *t* exceed 100 seconds by their imaginary parts, and all poles whose natural frequencies are less than 0.01 rads/sec by their real parts.

For the circuits with the time constants larger than 100 seconds, parameter ½**Re**½ has to be decreased.

For the circuits with extremely slow oscillations (<0,01 rads/sec), parameter ½**Im**½ has to be decreased.

2.2 Identification of multiple roots

Identification of multiple roots is necessary for the partial fraction expansion of the *s*-domain circuit function, and for the subsequent inverse Laplace transform for finding equations of the transient and impulse responses.

Criterion **r12**: identification of multiple roots.

In the first step, all roots are sorted into groups with "almost the same" absolute values: Absolute values in each group are equal in the frame of relative error **r12**. Inside each group, the "coincide test" is performed: both the real and imaginary parts have to be the same in the frame of relative error **r12**. If the roots are identified as the identical, their real and imaginary parts are replaced by the mean values.

Parameter **r12** must be neither too large nor too small. For a large **r12**, two different roots can be identified as multiple. For a small **r12**, multiple roots can be identified as different. This can be dangerous especially for the complex roots, when the algorithm need not find their conjugate counterparts. In such cases, parameter **r12** has to be increased.

Root polishing is finished by the algorithms which utilize parameters **m0** and **m9**.

2.3 Polishing chains of zeros and nines

Criterion **m0**: identification of a string of m0 zeros.

This root polishing method is explained in the following example:

before polishing	after polishing
-3.230000000047e-8	-3.23e-8

This algorithm tests the root's mantissa, whether a string of at least m0 zeros is present on the right side of the decimal point. If it is found, all numbers to the right of this string are replaced by zeros.

The longer the string of zeros the greater the probability that the subsequent nonzero numbers are due to rounding errors.

The permitted values of **m0** are integers from 2 to 13.

Criterion **m9**: identification of a string of m9 nines.

This root polishing method is explained in the following example:

before polishing	after polishing
-4.7999999999873e-4	-4.8e-4
0.9999999997389	1

This algorithm tests the root's mantissa, whether a string of at least m9 nines is present on the right of the decimal point. If it is found, this string is replaced by zeros including all subsequent numbers, and a prospective order transfer is performed to the left-side.

The longer the string of nines the greater the probability that it is due to rounding errors.

The permitted values of **m9** are integers from 2 to 13.

2.4 Zeros and poles cancellation

If "cancellation" is active, SNAP will try to cancel identical pairs of zeros and poles. The roots are considered to be identical, if both their real and imaginary parts are identical in the frame of the relative error **r12**.

3 Demonstration

Let us consider a cascade of two 2nd-order building blocks in Fig. 1. This filter is designed with the aim to reach so-called FIR-BL approximation of frequency response, which exhibits excellent group delay response [6]. Considering ideal operational amplifiers and the equalities

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

the quality factor of both blocks is 0.5 and the natural frequency is set to the value

$$w_0 = \frac{1}{2RC}\,.$$

The transfer function of each block is given by the equation

$$K_i(s) = a_i \frac{s^2 + w_{zi}^2}{(s + w_0)^2}\,,\ i = 1,\ 2,$$

where

$$a_i = \frac{R_{4i}}{R_{6i}}\,,\ w_{zi} = \frac{1}{RC}\sqrt{\frac{R_{6i}}{R}} = 2w_0\sqrt{\frac{R_{6i}}{R}}\,,\ i = 1,\ 2.$$

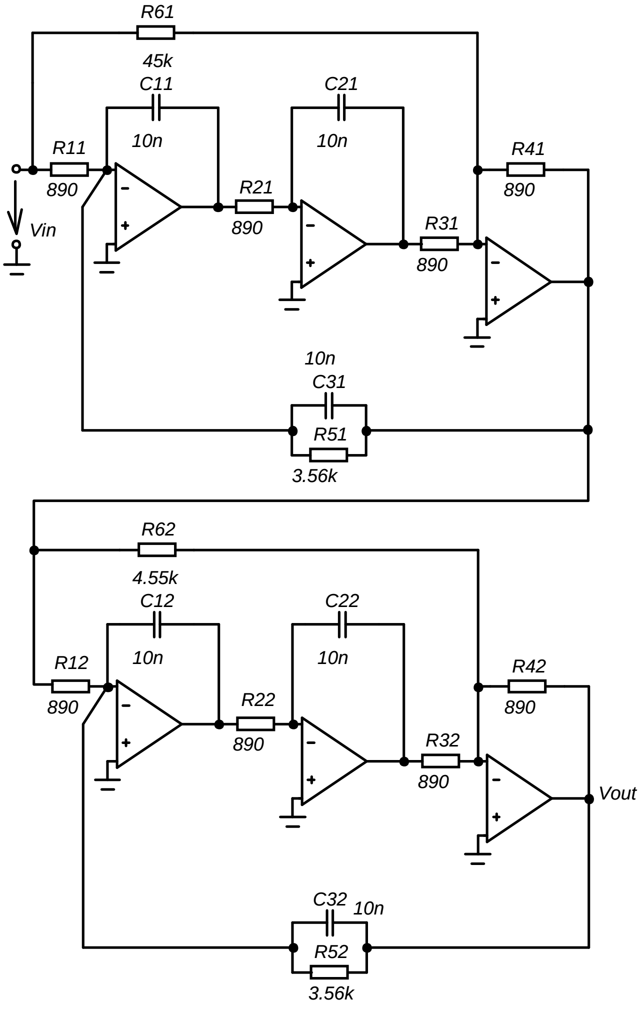

Fig. 1: An example of analyzed circuit.

That is why transfer function *Vout*/*Vin* of filter in Fig. 1 has two double zeros

$\pm jw_{z1} = \pm j7.98953027764.10^5$ rads/sec,
$\pm jw_{z2} = \pm j2.54050871091.10^5$ rads/sec,

and one quadruple pole

$-w_0 = -5.61797752809.10^4$ rads/sec.

Results of analysis by means of SNAP program are summarized in Table 1.

without secondary root polishing	
zeros	+j7.98953027764E5
	-j7.98953027764E5
	+j2.54050871091E5
	-j2.54050871091E5
poles	-5.61813401118E4 - j2.94208471184E-25
	-5.61797752547E4 + j1.62558164422E0
	-5.61797752525E4 - j1.56365534119E0
	-5.61780942729E4
h(*t*)	-1.60000331032E1 -1.84535365307E14*exp(-5.61813401118E4*t) +6.35334411613E12*exp(-5.61797752547E4*t) +6.59586418344E12*exp(-5.61797752525E4*t) +1.71586157008E14*exp(-5.61780942729E4*t)
with secondary root polishing	
zeros	+j7.98953027764E5
	-j7.98953027764E5
	+j2.54050871091E5
	-j2.54050871091E5
poles	-5.61797462230E4
	-5.61797462230E4
	-5.61797462230E4
	-5.61797462230E4
h(*t*)	1.60000331027E1 -2.99045155494E15/3!*t^3*exp(-5.61797462230E4*t) -4.77429680881E10/2!*t^2*exp(-5.61797462230E4*t) -8.99529813589E5*t*exp(-5.61797462230E4*t) -1.59961644825E1*exp(-5.61797462230E4*t)

Table 1: SNAP – results of the analysis of filter in Fig. 1; *h*(*t*) is step response.

Zeros and poles were computed from coefficient of numerator/denominator polynomials of symbolic transfer function after setting numerical values instead of symbolic parameters. In SNAP, roots of polynomials are found by specially modified Laguerre's method [3,4,5]. Without secondary root polishing, only zeros are found correctly. Instead of real quadruple pole, SNAP found four different poles, one real and three complex. That is why the consequential procedures of inverse Laplace transform for finding formula of step response will fail.

After enabling root polishing, SNAP will find four identical poles and procedure of partial fraction expansion as a part of the algorithm of inverse Laplace transform will generate correct results.

Fig. 2 and Fig. 3 show results of frequency and transient analyses of filter in Fig. 1. SNAP generates them on the basis of zeros and poles. It should be noted that for disabled root polishing only outputs of frequency analysis are practically applicable, whereas results of transient analysis are useless.

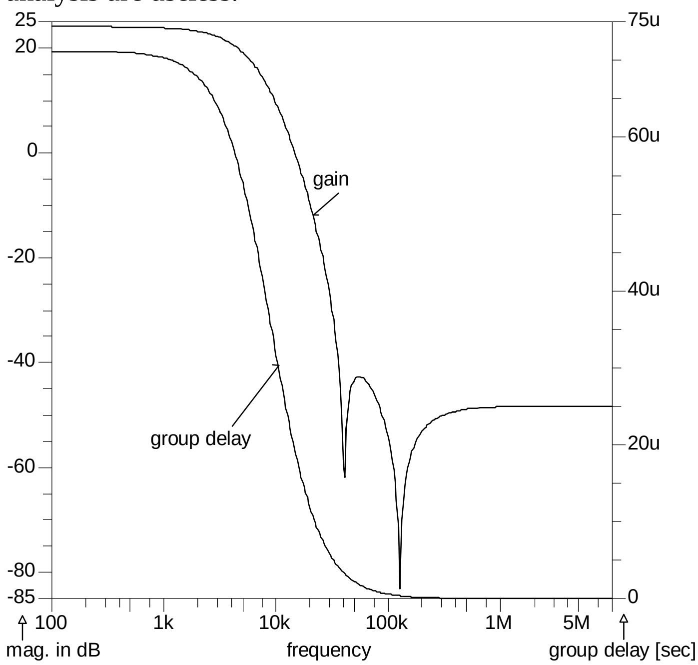

Fig. 2: Gain and group delay frequency responses, computed by SNAP program from zeros and poles. Results are practically the same before and after root polishing.

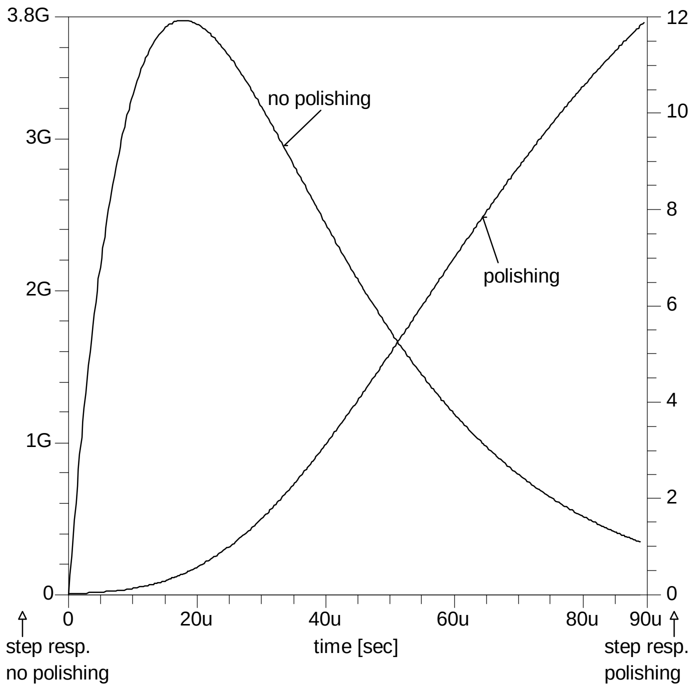

Fig. 3: Step response, computed by SNAP, without and with secondary root polishing. The response without polishing does not correspond to reality.

4 Conclusions

The described methods of secondary root polishing have proved to be a useful complement of the well-known primary root polishing techniques. They are especially important for the identification of multiple roots.

These techniques are included in the SNAP network analyzer [1]. Controlling constants of the individual algorithms are user-accessible in global settings menu. The default setting ensures a proper root polishing for a wide class of analyzed circuits.

It is also possible to extend the secondary root polishing techniques beyond the area of root computation, and to combine them with the primary polishing.

Acknowledgments

This work is supported by the Grant Agency of the Czech Republic under grants No. 102/03/1363 and 102/04/0442, and by the research programmes of BUT MSM 262200022 & MSM 262200011.

References:

[1] Biolek,D. SNAP - Program with Symbolic Core for Educational Purposes. Int. WSEAS Multiconference, *CSCC'00*, Vouliagmeni, Athens, 2000, pp. 1711-1714.

[2] Dobeš, J. An accurate poles-zeros analysis for large-scale analog and digital circuits. *IEEE Int. Conf. on Electronics, Circuits and Systems*, St. Julians, Malta 2001, pp. 1027-1030.

[3] Press,W.H. et al. Numerical Recipes in Pascal. The Art of Scientific Computing. *Cambridge University Press*, 1994.

[4] Acton, F.S. Numerical methods that usually work. *The Mathematical Association of America*, Washington D.C., 1995.

[5] Ralston,A. A First Course in Numerical Analysis. *McGraw-Hill Company*, 1965.

[6] Biolek,D.-Biolková,V. 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.