Accurate Semisymbolic Analysis of Circuits with Multiple Roots

ZDENEK KOLKA¹, MARTIN HORAK¹, DALIBOR BIOLEK², VIERA BIOLKOVA¹
¹Department of Radio Electronics, ²Department of Microelectronics
Brno University of Technology
Purkyňova 118, 612 00 Brno
CZECH REPUBLIC
kolka@feec.vutbr.cz

Abstract: The paper deals with a method for accurate computation of multiple poles and zeros in semisymbolic analysis of idealized linear circuits. The well known problem of the QR and QZ algorithms is their poor accuracy in case of multiple roots, which is usually compensated by the use of slow multiprecision arithmetic. The method presented in this paper is based on an improved reduction procedure for transforming generalized eigenproblem into the standard one in combination with an algorithm for computing the Jordan canonical form of inexact matrices [1]. The reduction procedure uses the SVD method for explicit rank estimation with the aim of avoiding the reporting of spurious roots. Numerical experiments have shown the numerical accuracy to be maintained even for defect matrices with high multiplicity roots.

Key-Words: Pole-zero analysis, Linear circuits, QR, QZ, Numerical methods

1 Introduction

The pole-zero analysis plays an important role in the design of electronic circuits. The computation 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 and defect matrices. As shown in [2], computational errors can have severe consequences for the semisymbolic analysis of pulse and step responses.

One of the most difficult problems in semisymbolic analysis is the computing of multiple roots. In fact, the question whether there exists a multiple eigenvalue is an ill-posed problem [4]. 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 depends on the accuracy of computed roots.

There is an attainable accuracy barrier in computing multiple roots [5]: 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 [6].

The pole-zero analysis of a linear circuit leads to the generalized eigenvalue problem, which can be solved using either the QZ algorithm [7] or it can be reduced algebraically to the standard one solved, for example, by the QR algorithm [8].

We will focus on the second approach using an enhanced reduction algorithm similar to [6]. Due to unavoidable numerical errors of floating point operations the pole-zero analysis often reports spurious roots. The proposed method uses the SVD method for explicit rank estimation with the aim of avoiding spurious roots.

Subsequently, the resulting standard eigenvalue problem is solved using a new algorithm for computing the Jordan canonical form of inexact matrices [1]. Unlike standard methods, which treat multiple roots as a collection of single ones, the algorithm [1] estimates the multiplicity structure first and then finds 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 reduction algorithm and Section 3 gives some numerical examples.

2 The Reduction Algorithm

Let $(A, B)$ be a matrix pencil whose eigenvalues are zeros or poles of a network function. The procedure for obtaining $(A, B)$ from circuit matrix can be found in [9]. Let us denote

$$C(s) = A - sB \quad (1)$$

Our goal is to transform the matrix pencil into a matrix $Q$ which has (approximately) the same eigenvalues in two steps. The first step converts the
matrix $C$ to form (2) using a procedure similar to the Gaussian elimination, and the second step reduces the matrix $C^{(0)}$ to form (3) by the pivotal condensation.

\[
C^{(0)}(s) = \begin{pmatrix} A_{11} - sI & A_{12} \\ A_{21} & A_{22} \end{pmatrix}
\]

(2)

\[
C^{(II)}(s) = Q - sI
\]

(3)

2.1 Step 1 – conversion of $C$ to $C^{(0)}$
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 with very big absolute value. To eliminate the effect, we compute the rank $r$ of $B$ beforehand by the Singular Value Decomposition (SVD), which is known as the most reliable algorithm for this task [8].

Performing only $r$ steps of the Gaussian elimination converts the upper-left part of matrix $B$ to the upper triangular form.

\[
a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - a_{k,j}^{(k-1)}b_{i,k}^{(k-1)}/b_{k,k}^{(k-1)} \quad k = 1,2,...,r
\]

(4)

\[
b_{i,j}^{(k)} = b_{i,j}^{(k-1)} - b_{k,j}^{(k-1)}a_{i,k}^{(k-1)}/b_{k,k}^{(k-1)} \quad j = 1,...,n
\]

An element $b_{i,j}$ with the maximum absolute value is chosen at each $k$-th step as the pivot $b_{k,k}$ from the rest of the matrix $B$

\[
b_{k,k} = \max_{i,k,j,k} \left| b_{i,j} \right|
\]

(5)

Formally, the $k$-th row and the $k$-th column are swapped with the $i$-th row and the $j$-th column of that element. This operation also ensures that the pivot element $b_{k,k}$ is always nonzero at $k$-th step.

Next, the upper-left part of matrix $B$ is converted to diagonal matrix using a transposed version of (4).

\[
a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - a_{i,k}^{(k-1)}b_{j,k}^{(k-1)}/b_{k,k}^{(k-1)} \quad i = k+1,...,n
\]

(6)

\[
b_{i,j}^{(k)} = b_{i,j}^{(k-1)} - b_{i,k}^{(k-1)}a_{j,k}^{(k-1)}/b_{k,k}^{(k-1)} \quad j = 1,...,n
\]

The last part of the conversion normalizes $r$ diagonal elements of $B$ to one by

\[
a_{i,j} = a_{i,j}/b_{j,j} \quad i = 1,2,...,r
\]

(7)

\[
b_{i,j} = 1 \quad j = 1,2,...,n
\]

All other nonzero elements of $B$ are considered as rounding errors, which can be zeroed.

2.2 Step 2 – conversion of $C^{(0)}$ to form $C^{(00)}$
An element $a_{k,j}$ with a maximum absolute value $a_{\text{max}}$ is chosen as a pivot from the submatrix $A_{22}$.

\[
a_{\text{max}} = \max_{k=r,a_{i,j},l=r,a_{i,j}} \left| a_{i,j} \right|
\]

(8)

a) If this pivot element has a (sufficiently) nonzero value, the pivotal condensation can be performed by

\[
a_{i,j} = a_{i,j} - a_{i,k}a_{j,k}/a_{k,k} \quad \{ i = 1,2,...,n \land i \neq k \}
\]

(9)

and then the $k$-th row and the $l$-th column are subsequently deleted from matrix $C$. Note that this operation reduces the order $n$ of the matrix $C$, but keeps the structure and rank of the associated matrix $B$, because $k$-th row and $l$-th column of the matrix $B$ are zero vectors.

b) If the pivot selected by (8) is zero, i.e. the submatrix $A_{22}$ is a zero matrix, the pivotal condensation cannot be performed with this pivot. If also both matrices $A_{12}$ and $A_{21}$ are zero matrices, this signalizes that the solved problem has an infinite number of eigenvalues. All rows and columns pertaining to the submatrix $A_{22}$ can be deleted from $C$.

If at least one of matrices $A_{12}$ or $A_{21}$ has a nonzero entry, the pivot $a_{i,j}$ is selected by

\[
a_{\text{max}} = \max_{k=r,a_{i,j},l=r,a_{i,j}} \left| a_{i,j} \right|
\]

(10)

The pivotal condensation by (10) can then be performed similarly as in the case a). Because the pivot is selected from matrix $A_{12}$ or $A_{21}$, this operation introduces additional nonzero elements into the matrix $B$. If the pivot is chosen from the matrix $A_{12}$, then matrix $B$ has the following form

\[
B_r = \begin{pmatrix} \text{I} & b^{(1)} & 0 & 0 \\ 0 & b^{(2)} & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}
\]

(11)

where

\[
b^{(1)} = \left[ b_{1,j}/a_{k,j}, b_{2,j}/a_{k,j}, ..., b_{k-1,j}/a_{k,j}, b_{k,j}/a_{k,j} \right]^T
\]

(12)

\[
b^{(2)} = \left[ b_{k+1,j}/a_{k,j}, b_{k+2,j}/a_{k,j}, ..., b_r,j/a_{k,j} \right]^T
\]

(13)

Now it is possible to zero-out the newly introduced nonzero elements in matrix $B$ and update matrix $A$ by

\[
a_{i,k} = a_{i,k} - \sum_{j=1,j \neq k} a_{i,j}b_{j,k} \quad i = 1,2,...,n
\]

(14)

And finally, the $k$-th and $n$-th columns will be
exchanged in \( C \) to get the form \( C^{(II)} \). The same procedure holds if the pivot is chosen from \( A_{21} \), but it is necessary to utilize the transposed versions of (12) – (14).

Note that the case \( a) \) reduces only the order \( n \), the case \( b) \) also reduces both the rank \( r \) and the order \( n \) of matrix \( B \). However, in all cases, matrix \( C \) is converted to form (2). The above steps should be performed until the order of the matrix \( A_{22} \) is zero.

The reduction algorithm takes the linear matrix pencil and transforms it into a single matrix, \( Q \), which is a part of the \( ApaTools \) package [3]. First, the procedure estimates the eigenvalue multiplicity structure (or equivalently finds Jordan blocks) and then computes eigenvalues using a well-conditioned iteration process [1]. Both the reduction algorithm and \( ApaTools \) were implemented in Matlab.

3 Numerical Examples

The proposed method (referred to as \( RedJo \)) was tested in Matlab (v7.8.0.347). The results were compared with the Matlab \( QZ \) function and with HSpice and HSpice RF (v2007.09).

Figure 1 shows a two-stage filter designed using the FIR-BL approximation [2]. Considering the ideal operational amplifiers and the following equalities

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

the transfer function of each stage is

\[
K_i(s) = a_i \frac{s^2 + \omega_0^2}{(s + \omega_0)^3}, \quad i = 1, 2 \quad (15)
\]

where

\[
a_i = \frac{R_{4i}}{R_{6i}}, \quad \omega_{2i} = \frac{1}{RC} \sqrt{\frac{R_{6i}}{R}} \quad i = 1, 2 \quad \omega_0 = \frac{1}{2RC} \quad (16)
\]

The transfer function has two double zeros and one quadruple pole

\[
z_{1,2} = \pm j7.98953027764 \times 10^5 \\
z_{3,4} = \pm j2.54005871091 \times 10^5 \\
\rho_{1,2,3,4} = -5.61797752809 \times 10^4
\]

(17)

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. The simulation circuit for HSpice was based on the built-in model of the ideal operational amplifier. Although the \( PZ \) analysis in HSpice failed completely, the results of the AC analysis were correct. Table I shows the results of the algorithms tested using standard IEEE 754 double-precision arithmetic. \( RedJo \) found correctly both the pole and its multiplicity

![Fig. 1 Idealized two-stage active filter.](image_url)

<table>
<thead>
<tr>
<th>( \text{RedJo} )</th>
<th>(-5.61797752809898 \times 10^4 ) (quadruple pole)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Matlab QZ} )</td>
<td>(-5.617977528098 \times 10^4 )</td>
</tr>
<tr>
<td></td>
<td>(-5.617977528098 \times 10^4 )</td>
</tr>
<tr>
<td>( \text{HSpice} )</td>
<td>(-5.617977528098 \times 10^4 )</td>
</tr>
<tr>
<td></td>
<td>(-5.617977528098 \times 10^4 )</td>
</tr>
<tr>
<td>( \text{HSpiceRF} )</td>
<td>0.00000 (triple pole)</td>
</tr>
<tr>
<td></td>
<td>(-4.5625 \times 10^{12} )</td>
</tr>
<tr>
<td></td>
<td>(-2.4825 \times 10^{8} )</td>
</tr>
<tr>
<td></td>
<td>(-1.4045 \times 10^{3} )</td>
</tr>
</tbody>
</table>

Table II shows the results of pole analysis for different numbers of sections obtained using the \( QZ \) algorithm, the reduction and \( QR \) algorithm (\( RedQR \)) and the reduction with approximate Jordan decomposition (\( RedJo \)). The exact pole value was computed using (16).

In the ideal case, for \( n \) sections we obtain a \( 2n \)-fold pole. As \( QZ \) and \( QR \) algorithms provide only a cluster of poles, the error shown is the maximum relative distance from \(-\omega_0\), \( \max\|\rho_i + \omega_0\| \). Algorithm
RedJo found correctly the pole multiplicity in all cases providing accuracy comparable to the machine precision. Both QR and QZ lose their precision quickly for higher multiplicities. The poor performance of the RedQR algorithm is caused by the accumulation of errors during matrix reduction.

<table>
<thead>
<tr>
<th>n</th>
<th>QR</th>
<th>RedQR</th>
<th>RedJo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$2.6 \times 10^{-16}$</td>
<td>$3.2 \times 10^{-16}$</td>
<td>$3.9 \times 10^{-16}$</td>
</tr>
<tr>
<td>4</td>
<td>$4.0 \times 10^{-8}$</td>
<td>$1.7 \times 10^{-8}$</td>
<td>$0.0$</td>
</tr>
<tr>
<td>100</td>
<td>$0.3 \times 10^{-2}$</td>
<td>$0.3 \times 10^{-2}$</td>
<td>$0.0$</td>
</tr>
</tbody>
</table>

Table II Maximum relative error of computed poles

Figure 2 shows the ability of RedJo to separate nearby multiple roots. The circuit in Fig. 1 is analyzed with different values of capacitors

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

The value of $C_1$ is firmly set to 10nF, while the value of $C_2$ is swept from 0.9985 to 1.0015 of the nominal value of 10nF. The poles are then

$$p_{1,2} = -\frac{1}{2RC_1}, \quad p_{3,4} = -\frac{1}{2RC_2}.$$ (18)

Theoretically, it is only for $C_2 = 10$ nF that the circuit has a quadruple pole. For other values it has two double poles. 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.

4 Conclusions

The proposed method is suitable for the analysis of idealized circuits where multiple eigenvalues and matrix defectiveness are likely to occur. Multiple roots are computed reliably even with standard IEEE 754 double-precision arithmetic.

5 Acknowledgements

This work was supported by the Czech Science Foundation under projects No. 102/08/0851, No. 102/08/0784, and by the Czech Ministry of Education under research projects No. MSM0021630513 and OC09016, which is a part of the COST Action IC0803 RFCSET. The research has also received funding from the European Community’s Seventh Framework Programme under grant agreement No. 230126.

References:


