SNAP - PROGRAM FOR SYMBOLIC ANALYSIS

Zdeněk KOLKA
Dept. of Radioelectronics
Technical University of Brno
Purkyňova 118, 612 00 Brno
Czech Republic
kolka@fee.vutbr.cz

Abstract

The paper deals with a program SNAP for symbolic analysis of linear circuits in frequency domain. The program is suitable for analysis of circuits with ideal network elements to explore basic principles of their operation. Besides graphical presentation the analysis results can be exported to popular mathematical programs for further processing. Currently, an algorithm for exact symbolic analysis is implemented. Therefore the program is suitable for relatively small circuits.

Keywords

circuit theory, symbolic analysis, CAD

1. Introduction

The program SNAP is a successor of popular COCO program. It has been developed with the features:

- The circuit analyzed is read from a netlist file. Any universal schematic editor can be used in conjunction with SNAP.
- Models of network elements are stored outside the program in an easily extensible library.
- The program performs also numeric analysis.

2. Algorithms

The linear circuit analyzed is described by the modified generalized nodal analysis [2] with hybrid network matrix **H**. Each network element is described by its stamp, i.e. by a special matrix indicating the position of parameters and coefficients in the resulting matrix **H**. The stamp stored in a library defines a set of rules for transformation of network element parameters from netlist into the matrix **H**. The particular parameter can simultaneously appear on four or less positions in **H**. This is described by structure

$$\boxed{P}\,\boxed{a\,|\,b\,|\,c\,|\,d} \tag{1}$$

called an *atom*. The parameter P appears on positions (a, b) and (c, d) with positive sign and on (a, d) and (c, b) with negative sign. Index 0 represents position outside of **H**. The P from (1) is multiplied with another P's during determinant expansion. Thus without expense it can be more generally defined as

$$P = k\,s^m \prod_i r_i^{n_i} \tag{2}$$

where k , m are constants, s is Laplace operator, and r_i denotes i-th parameter of a particular network element with positive or negative power n_i . Fig. 1 shows model of a converter and its atoms. They are stored in the program library that can easily be modified or extended.

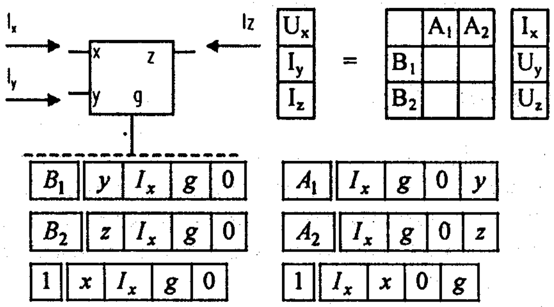


Fig. 1 Three-port converter, its matrix, and atoms

Circuit functions are computed using algebraic cofactors of **H**. The whole circuit is described by a set of atoms (1), so the matrix **H** is not in fact necessary. The determinant expansion procedure is based on the recursive formula [2]

$$\Delta = \tilde{\Delta} + P_i\,\tilde{\Delta}_{a:b}^{[a>c:b>d]} \tag{3}$$

where Δ is determinant of a matrix, $\tilde{\Delta}$ and $\tilde{\Delta}_{a:b}^{[a>c:b>d]}$ are determinant and algebraic cofactor of the matrix without atom $\boxed{P_i}\,\boxed{a\,|\,b\,|\,c\,|\,d}$. The same rule is applied to $\tilde{\Delta}$ and $\tilde{\Delta}_{a:b}^{[a>c:b>d]}$. The expansion of a particular branch stops successfully when a determinant of a zero-dimension matrix is to be computed. If it stops from another reason (empty matrix, nonexistence of cofactor) then branch is discharged. The algorithm has been enhanced by graph-theory based tests that can detect unsuccessful branches to decrease number of steps performed [1].

3. Postprocessing

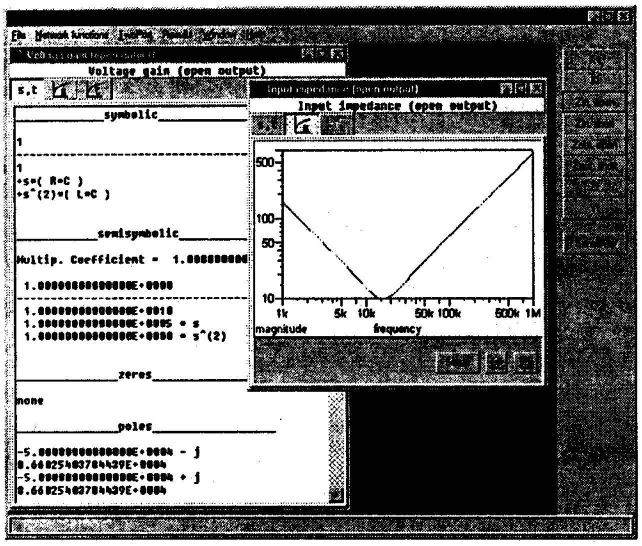
All circuit functions obtained in symbolic form are simplified, if possible. The program can also compute absolute and relative sensitivities by means of symbolic differentiation of network functions.

Parameters of circuit elements can be defined in netlist symbolically, numerically or as a combination of both. If all parameters are given a numeric value then postprocessing can continue in the numeric domain. Poles and zeros are determined using Laguer's method from the polynomials. The method gives satisfactory results for circuits whose dimension is reasonable from symbolic analysis viewpoint. Then inverse Laplace transform (ILT) can easily be performed to obtain pulse and step responses of the circuit in semi-symbolic form. A method of secondary root polishing is applied to the poles and zeros [3]. It is useful when round off errors can qualitatively change results of ILT. For example, two poles [-1, -0.99999999999] are obtained. It is probable the correct answer was [-1, -1]. The tiny numeric error causes the result of ILT qualitatively differ from the correct one. The method searches for this and similar situations and tries to eliminate the inevitable round off errors.

4. User interface

The user interface was designed according to present-day standards for MS Windows programs. SNAP can operate in conjunction with any schematic editor producing appropriate netlist. Currently it is used with Schematics from PSpice package.

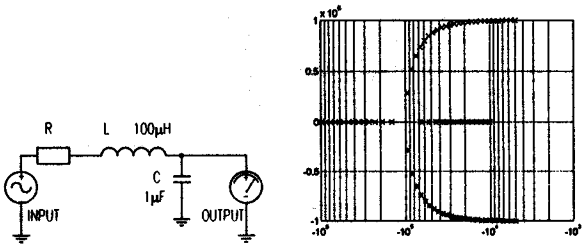
Numerical results can be visualized in a graph which can be saved or copied in a vector format. Any circuit parameter can be stepped and additional dependences between circuit variables can be introduced.



```
_____________symbolic____________
1
---------------------------------
1
+s*( R*C )
+s^(2)*( L*C )

___________semisymbolic__________
Multip. Coefficient =  1.00000000
 1.00000000000000E+0000
---------------------------------
 1.00000000000000E+0010
 1.00000000000000E+0005 * s
 1.00000000000000E+0000 * s^(2)

______________zeros______________
none

______________poles__________________
-5.00000000000000E+0004 - j
8.66025403784439E+0004
-5.00000000000000E+0004 + j
8.66025403784439E+0004
```

Fig. 2 SNAP user interface

Analysis results can be exported to popular mathematical programs Matlab, Maple, and MathCAD for further processing. Results saved in a compact Matlab function can be called directly in Matlab. The function allows to compute its numeric value for a given *s*, to obtain coefficients of numerator and denominator, and to work with symbols for circuit variables. Fig. 3 shows simple script for root-locus diagram. The function *Ku_RLC()* was generated by SNAP.



```
Ku_RLC('export'); global R;              %binding of variable R into Ku_RLC
N=40; Rstep=logdiv(1,100,N);             %logarithmic division of R
for i=1:N,                               %roots
  R=Rstep(i); Ro(1:2,i)=roots(Ku_RLC('denom'));
end;
                                         %graph
semilogx(real(Ro(1,:)),imag(Ro(1,:)),'wx',real(Ro(2,:)),imag(Ro(2,:)),'wx');
grid on;
```

Fig. 3 Example of root-locus diagram generation in Matlab

5. Conclusion

SNAP was found useful for circuit theory education as well as for basic circuit analysis where eliminates need for time consuming hand calculations. Because of exact symbolic analysis used the circuit size is limited to several nodes. The next generation SNAP will incorporate derivation of linearized circuit parameters directly from a standard Spice netlist as well as approximating symbolic analysis.

References

[1] TABERY,M.: Program for semisymbolic analysis of linear circuits. Master's thesis. Technical University Brno, 1996.

[2] ČAJKA,J.-KVASIL,J.: Theory of linear circuits. SNTL Prague, 1979.

[3] BIOLEK,D.: Secondary root polishing techniques in the computer circuit analysis. In: *Proc. „Radioelektronika'99"*. (will be published)

About author...

Zdeněk KOLKA was born in Brno, Czechoslovakia, in 1969. He received the M.S. (92) and Ph.D. (97) degrees in electrical engineering, both from the Faculty of Electrical Engineering and Computer Science, Technical University of Brno. At present he is an Assistant Professor at the Department of Radioelectronics, Tech. Univ. of Brno. He is interested in PWL modeling, circuit simulation, and nonlinear dynamic systems.