SNAP PROGRAM EXTENSION FOR SUPPORTING USER-BASED ANALYSIS AND OPTIMIZATION OF ANALOG FILTERS

DALIBOR BIOLEK[*)], **ZDENĚK KOLKA**[**)]

Abstract: *New features of the SNAP program for analysis of linear and linearized circuits are described in this article. These features have been implemented to make user-based analysis and optimization of frequency filters more effective. Students can currently work with the program both during the training and by working on individual projects.*

Key words. Symbolic and semisymbolic analysis, netlist, MATLAB

1. Introduction

SNAP program (Symbolic Network Analysis Program) has been originally compiled for the symbolic analysis of general linear and linearized circuits. This window-based analyzer uses the symbolic algorithms by Prof. Čajka. Step by step, this program has been extended to the semisymbolic and numeric analyses and the core of symbolic computation was improved with the utilization of the graph theory.

The circuit scheme is initially created in a schematic editor where it is converted automatically to the *snap netlist* (a file with the .snn extension). After that, the SNAP program is started. SNAP reads the netlist data and a corresponding analysis is performed.

At the present time, a large program reconstruction is in progress. All the program modifications are inspired partly by the needs of computer supported design, partly they are required by current users and students.

In this paper, we focus on the following innovations:

Linkages in the symbolic analysis.
Linkages in the numeric analysis.
Wave Clipboard.
Results export to the universal computing programs.

2. Linkages in the symbolic analysis

In the netlist, each circuit element is defined in an individual line in the standard SPICE format. The schematic editor generates all this automatically. An example of resistor definition is given below:

R_R1 1 2 18k ... resistor, label R1, connected between nodes 1 and 2, its resistance is 18kΩ.

The parameter which terminates the line can be defined in five various ways:

[*)] MA Brno, K301, Kounicova 65, PS13, 612 00 Brno, tel. (05)41182487, email biolek@cs.vabo.cz
[**)] IREL BUT Brno, Purkyňova 118, 612 00 Brno, tel. (05)41149148, email kolka@urel.fee.vutbr.cz

Example:	in general:	
1) R1	*symbol*	
2) 18k	*value*	
3) R1:18k	*symbol:value*	Note: methods 3 and 4 are equivalent
4) R1=18k	*symbol=value*	
*5) 2*R3*	*number followed by operator (* / + -), symbol [, operator, number]*	

Here

symbol symbolic name of the parameter

value numeric value of the parameter according to the Pspice standard.

The specified method of parameter definition then impacts the analysis type in the SNAP program:

To enable semisymbolic and numeric analyses, numeric information about all parameters of all components has to be included in the netlist (i.e. articles 2, 3 or 4). If a numeric value is not known, only symbolic results will be available.

Moreover, a proper parameter definition can lead to a simplification of symbolic results. As a typical example, we can label several resistors of the same resistance with a single symbol.

Example: The Wien cell is composed of identical resistors R1=R2=1kΩ and capacitors C1=C2=10nF.

R_R1 1 2 R1=1k	resistor, resistance R1=1kΩ, connected between nodes 1 and 2
C_C1 2 3 C1=10n	capacitor, capacitance C1=10nF, between nodes 2 and 3
R_R2 3 4 R1	resistor, resistance R1 (=1kΩ), connected between nodes 3 and 4
C_C2 3 4 C1	capacitor, capacitance C1 (=10nF), between nodes 3 and 4
I_I1 1 4	input gate between nodes 1 and 4
O_O1 3 4	output gate between nodes 3 and 4

The symbolic result is already simplified, taking into account the given R and C equalities:

$$K_V = \frac{s * R_1 C_1}{1 + s * 3R_1 C_1 + s^2 * R_1^2 C_1^2}$$

3. Linkages in the numeric analysis

Upon the frequency filter analysis, it is advantageous to consider linkages among the parameters of circuit elements and the filter parameters. In the active 2nd order filter, for example, R1 is necessary to be stepped such that the quality factor remains constant. Then while R1 is being stepped, some other parameters have to be varied to fulfill this linkage condition.

Manifold stepping as in the MicroCap 5 program seems to be a possible solution. However, the introduction of additional analysis conditions can be preferable. For these purposes, SNAP offers the „dependences“ window for entering the linkages. They are evaluated in the same sequence in which they were entered.

Example: *2nd order Sallen-Key lowpass filter with the ideal OpAmp* is defined by the netlist:

```
OPA_O1  1  2  2  3    Operational Amplifier (OpAmp): input, input, output, common outlet
R_R1  4  5 R1=10k
R_R2  5  1 R2=10k
C_C1  5  2 C1=100n
C_C2  1  3 C2=22n
I_I1  4  3
O_O2  2  3
```

Symbolic and semisymbolic results:

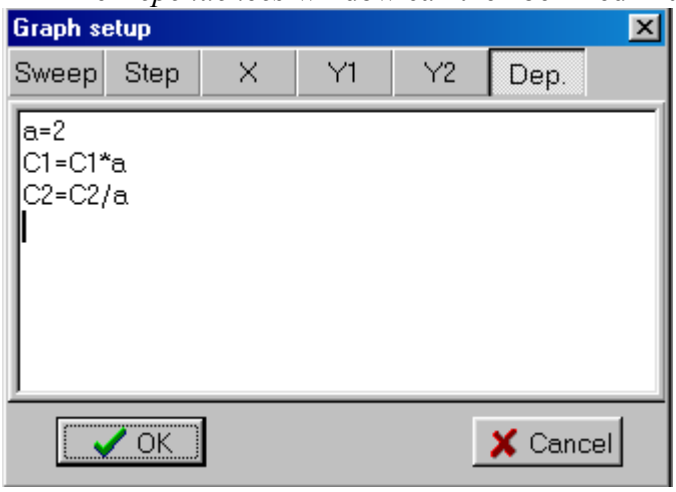
$$\frac{1}{1+sC_2(R_1+R_2)+s^2R_1R_2C_1C_2}=\frac{4.54545454545454.10^6}{4.54545454545454.10^6+2000s+s^2}.$$

The filter is characterized by frequency ω_0 or f_0, and by quality factor Q:

$$f_0=\frac{1}{2\pi\sqrt{R_1R_2C_1C_2}}\doteq 339\text{ Hz},\ Q=\sqrt{\frac{C_1}{C_2}}\frac{\sqrt{R_1R_2}}{R_1+R_2}\doteq 1.$$

For example, if we intend to increase Q a-times while keeping f_0 constant, we can achieve this by increasing C_1 a-times while simultaneously decreasing C_2 a-times.

The *Dependences* window can then be filled in as shown:



In the first line, we introduced variable a, setting its value to 2. The second/third line means a twofold increase/decrease in C1/C2 compared to the nominal values.

Now the analysis will be performed for values C1=200nF and C2=11nF. Frequency f_0 will remain unchanged, but the quality factor is now 2.

Defining variable a in the first line enables its stepping. In the *Step* folder, this variable appears among the offered circuit parameters that can be stepped.

4. Wave Clipboard

The Wave Clipboard can be profitably used to save curves to a single graph window. In this way, one can compare analysis results under different conditions, e.g. for various levels of the component models, for the circuit modification by additional elements and parasitics, etc.

The basic limitation results from the main purpose of the Wave clipboard, i.e. “clip” of curves in a single graph: only curves of one type of circuit function can be stored in the Wave Clipboard. Filling the Clipboard with the curves of voltage gain, it is not possible to add the impedance curves, etc.

Except for this limitation, the curve copy to the Clipboard is resolved as generally as possible, because not only a curve but also complete symbolic and semisymbolic equations are copied. That is why we can additionally modify parameters of the curves saved in the Clipboard (e.g. it is possible to switch from frequency to transient analysis, etc.).

While copying a curve from the current graphic window to the Wave Clipboard, we are called to enter the curve name, which then becomes its identifier.

The Wave Clipboard window has practically the same attributes and features as the classical SNAP graphic window, including the scanning of curve coordinates by cursors, copying to the Windows clipboard, etc. Furthermore, the Wave Clipboard can be not only filled in, but we can also remove sequentially individual curves from it.

5. Export to other programs

The analysis results can be exported to the MATLAB, MAPLE, and MathCad programs for subsequent processing. What is essential is that you can then operate directly with the symbolic equations which were originally generated by SNAP. We present export to MATLAB.

Consider the aforementioned Wien cell. Let us perform the analysis of the voltage gain and then export the results to the WIEN.M file. We move this file to any folder in the MATLAB datapath. Entering

```
help wien
```

from the MATLAB command prompt yields the following action:

```
Voltage gain (open output) of circuit wien.snn
usage:      wien(s) - return complex value of circuit function for given s
            wien('numer')  - return vector of numerator coefficients
            wien('denom')  - return vector of denominator coefficients
            wien('export') - export network parameters to global workspace
            wien('show')   - show network parameters
            wien('clear')  - clear global workspace
polynomials are in the Matlab style, i.e. C(1)*s^N + ... + C(N)*s + C(N+1)
```

The next commands lead to the computation of the vectors of numerator (*b*) and denominator (*a*) coefficients of the voltage gain:

```
b=wien('numer')
b =
  1.0000e-005          0
a=wien('denom')
a =
1.0000e-010          3.0000e-005       1.0000e+000
```

Using the *a* and *b* vectors, one can perform an arbitrary circuit analysis with the help of MATLAB functions. For instance, the frequency response is plotted using the command

```
freqs(b,a)
```

In MATLAB, however, we can directly operate with the variables which represent parameters of individual circuit elements (R1 and C1 in our case). For this purpose, it is necessary to export these parameters to the global parameter workspace, using the command

```
wien('export')
```

For example, analyze a circuit for various values of R1. We use the standard MATLAB command

```
global R1
```

Though variable R1 is defined as a local variable in the function inside the WIEN.M file, it is now accessible also from the command prompt. Let us modify R1 to 100Ω:

```
R1=100;
```

The numerator and denominator coefficients of the circuit function will now be changed according to the symbolic equations inside the WIEN.M file:

```
b=wien('numer')
b =
  1.0000e-006          0
a=wien('denom')
a =
1.0000e-012          3.0000e-006       1.0000e+000
```

6. References

[1] BIOLEK,D.-KOLKA,Z.: SNAP: A tool for the analysis and optimization of analogue filters. Submitted to ECCTD'99, Italy.

***Acknowledgement**: This work is supported by the GACR under grant No. 102/97/0765.*