

SAPWIN—A Symbolic Simulator as a Support in Electrical Engineering Education

Antonio Luchetta, *Member, IEEE*, Stefano Manetti, *Member, IEEE*, and Alberto Reatti

Index Terms—Computer aided design (CAD), electrical engineering, symbolic analysis.

I. SUMMARY

The use of computer based tools in electrical engineering courses should be always accompanied with the possibility to gain an insight into the circuit properties. This can be achieved by using a computer program which combines symbolic and numerical simulation capabilities. This approach in providing students of automatic analysis tools is more significant and complete. Therefore the program SAPWIN has been developed by the authors. This program includes the capability of generating the complete or approximate symbolic (fully or partially) output expressions of analog linear circuits and their graphical manipulation. In addition, it permits to obtain Spice netlist of a circuit, in order to work in a complete, self-sufficient environment, useful for both teachers and students. The program can be used “as is,” like an useful tool during basic electrical sciences or network theory lessons, or like a starting platform on which to develop further application that requires symbolic formulas, eventually directly operating on the program code. The positive outcome of the first release of Sapwin, here presented, urged us to work on a completely renewed one, which will be ready as soon as possible.

Antonio Luchetta (M'96) received the degree in Electronics Engineering from University of Florence, Italy, in 1993. He is a Researcher in the Department of Engineering and Environmental Physics at the University of Basilicata, Italy, where he also teaches electrical sciences. His research interests include software and algorithms for electric network simulation, neural network application to forecasting and control problem, elaboration of meteorological data. He is Member of IEEE and AEI.

Stefano Manetti (M'96) received the degree in electronic engineering from the University of Florence, Italy, in 1977. From 1977 to 1979 he had a research fellowship at the Engineering Faculty of Florence University. He was assistant professor of Applied Electronics at the Accademia Navale of Livorno from 1980 to 1983, and researcher at the Electronic Engineering Department of Florence University from 1983 to 1987. Since May 1987, he has been associate professor of Network Theory at the same University. In 1994, he join the University of Basilicata, in Potenza, Italy, as a full professor of Electrical Sciences. Since November 1996, he is full professor of Electrical Sciences at the University of Florence. His research interests are in the fields of circuit theory, neural networks and fault diagnosis of electronic circuits. He is Member of IEEE, ECS and AEI.

Alberto Reatti received the degree in Electronics Engineering in 1988 and the Ph.D. degree in 1993, both from the University of Florence, Italy. Since 1992, he has been Associate Researcher with the department of Electric Engineering, Wright State University, Dayton, OH. In 1994, he was Visiting Professor with the Department of Electronics at the University of Siena. He was also Researcher at the Department of Electronics and Telecommunications at the University of Florence where is currently working as an Associate Professor. His interests are in the electric network simulation and high-frequency high-efficiency power converters.

CD-ROM folder 12.

A. Luchetta is with the University of Basilicata, Contrada Macchia Romana Potenza, Italy.

S. Manetti and A. Reatti are with the University of Florence, Firenze, Italy.

Publisher Item Identifier S 0018-9359(01)05715-6.



©2001 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the [IEEE](http://www.ieee.org).

[Return to Table of Contents](#)

SAPWIN - A Symbolic Simulator as a Support in Electrical Engineering Education

A. Luchetta, *Member, IEEE*, S. Manetti, *Member, IEEE*, A. Reatti.

Abstract - Gaining an insight into circuit properties in electrical engineering classes can be achieved by using computer based tools. A computer program which combines symbolic and numerical simulation capabilities is of great help, because such a program provides students with automatic analysis tools. This paper presents the program [SAPWIN](#), which has been developed to perform an automatic symbolic and numerical analysis of linear circuits. The paper presents program features, their development lines and fundamental aspects. Also, the educational purposes which are contained in the use of program itself are presented.

I. Introduction

One of the main problems encountered in teaching modern electrical engineering is to individuate the tools which can better help the students in deeply understanding the concepts in a short time. Students can take some advantage of using “powerful” analysis/simulation tools since these tools result in very short calculation times. However, by using these tools, students solve problems without knowing “how” or “why” and they do not get any introspective ideas about the problem. One possibility is to leave the learner alone with problems and mathematical methods and let him solve these problems with the aid of only simple tools, e.g., only a pocket computer. This allows them to deeply understand every step of procedure but results in a tremendous time expense. However, none can claim that this is really useful, because there is a large availability of simulation and automatic calculation tools which can reduce the time expense in solving electrical engineering problems.

An appropriate solution consists of automating the most repetitive or trivial steps as much as possible without losing sight of the procedural technique. This is reached by employing a combination of numerical and symbolic simulators. Since at the time SAPWIN was introduced no similar tools were available, the authors developed this program in order to provide students with the visual impact of symbolic results while also introducing them to powerful features of a numerical program, such as SPICE. We tried to provide students with a tool capable of giving both conclusive results and a deep knowledge of the circuit behavior. This has been achieved by using symbolic techniques, which are widely recognized to be better suited for educational purposes than numerical procedures [\[1\]](#), [\[2\]](#).

Actually, symbolic algorithms provide to the users with the opportunity to face a wide spectrum of applications (evaluation of testability, fault diagnosis, device modeling, parameter identification,

nonlinear simulation) [3], [4]. Moreover, a symbolic simulator allows for an adequate description of circuit theory and makes easier and faster the understanding of some concepts, e.g., behavior of circuits containing an inductor cut or capacitor mesh or location of parameters which do not appear in output expressions. This is not achieved by using many powerful analog and digital simulators, since basic aspects of electrical sciences and specifically, circuit theory, are not considered when running these simulators. The student can only verify that results provided by the program agree with expectations, performing a numerical or graphical comparison between manually and automatically obtained results, because these programs do not help the user in understanding how parameter variations affect a given circuit operation.

The teaching of circuit analysis concepts and basic design techniques in network theory courses do not benefit from using classical programs, which are useful for expert designers, but not for inexperienced undergraduate students who still have difficulty in comprehending the concepts of stability, root position on the complex field, or optimization of sensitivity. On the other hand, the application of general purpose mathematical programs with symbolic capabilities, such as Mathematica or Maple, which are well-suited in electrical engineering courses [2], is limited by some drawbacks: a new language must be learned (this could be a good thing, but often requires too much time and effort with respect to course schedule); these programs are big and costly software packages requiring economical efforts to be run at home on a student PC. Actually, both traditional numerical simulators and advanced mathematical programs seem to be more suitable for expert designers, than for inexperienced undergraduate students.

Symbolic outputs resulting from the analysis of a circuit contain a higher information rate, however they may easily lead, even if the circuit is relatively simple, to extremely large symbolic expressions, which are very difficult to manage. By providing the symbolic simulation program with some approximation methods allows the user to take into account only the most important terms of the final expressions. Moreover, the circuit components most affecting the operation of a circuit can be individuated and considered. This feature has been introduced in the symbolic program illustrated in this paper.

II. The program package

In the past few years, the authors worked on a self-sufficient PC tool (this tool does not need external mathematical engines, like other software packages with similar scopes). This program is called [SAPWIN 1.0-1.2](#) and has been derived from a simpler DOS version called SAPEC. [SAPWIN](#) is a Windows program package for symbolic and numerical simulation of analog circuits. It is constituted by three modules: SAPWIN, SAPEC, and WINGRF, running interactively or independent of each other. SAPWIN allows the circuit scheme to be drawn and commands to be entered; SAPEC performs the circuit symbolic simulation, and WINGRF graphically processes the results of an already analyzed circuit.

The first version of [SAPWIN](#)[5], [6], has been used in classes on electric engineering theory and it has been used to develop further applications using its symbolic engine. It has also been delivered to many university-level teachers around the world. Good feedback about the qualities and faults, and many suggestions for possible improvements of the program have been returned to the authors. At present, this program is under further development and is continuously updated and enhanced to get the best performance with the most recent level variations of hardware. For more on this, see the [last paragraph](#).

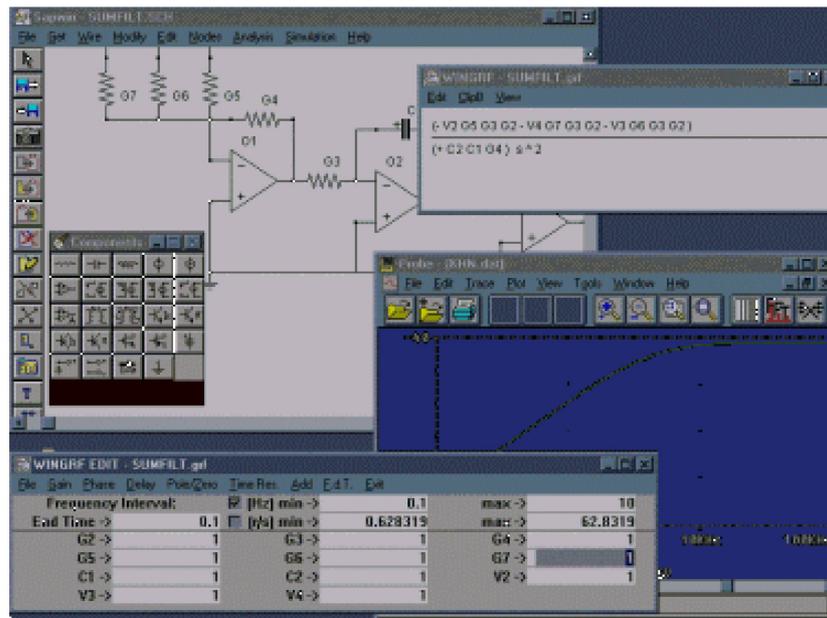


Fig. 1. Sapwin Ver.1.2 program environment.

A. Schematic entry

SAPWIN also includes a graphical schematic entry module. The main screen is a white sheet where the user can draw a circuit. All typical Windows tools to copy, cut, paste, move, edit a component or a part of the scheme are available. Figure 2 shows a schematic circuit entry when the Get and Wire options for the circuit are activated.

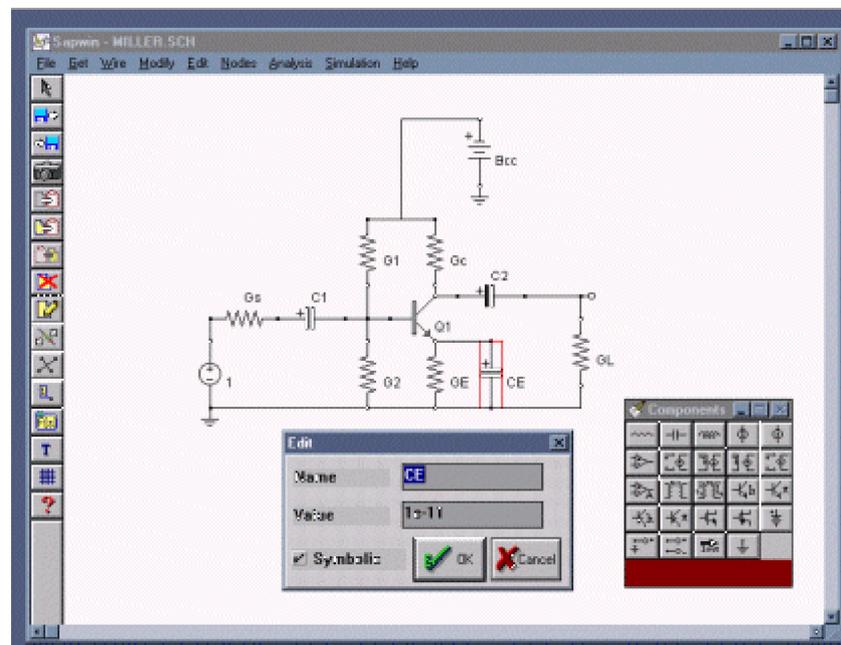


Fig. 2. Sapwin Ver.1.2 schematic editor.

The components which can be drawn by using the Get option are:

conductance;

- capacitor;
- inductor;
- independent voltage and current sources;
- the four kinds of controlled sources;
- mutual inductance;
- ideal transformer;
- ideal (nullor) and finite gain operational amplifier;
- semiconductor devices (bipolar and MOS transistors n-p-n or p-n-p type).

For semiconductor devices, linearized models (hybrid two-port parameters and Giacoletto) are available. Moreover, the user can select the parasitic components to be taken into account in the circuit analysis, such as the parasitic capacitors of a BJT and of a MOS transistor. The component values are assigned with the Edit option, while the Node option sets the circuit nodes. The File option allows the user to select an already drawn circuit. The symbolic analysis starts by entering the "Analysis" option. [SAPWIN](#) can also generate the circuit netlist file for an external numerical simulator. This file is a text file suitable for SPICE and the circuit analysis with this numerical simulator can be automatically activated inside SAPWIN. Moreover, the desired commands required by Spice netlists can be directly edited. The chance of using both symbolic and numerical analysis is very useful for circuit design purposes. The influence of both topology and component value change on circuit behavior can be immediately evaluated by means of symbolic simulation. On the other hand, the choice of component values and the final behavior can be accurately verified by means of a reliable numerical simulator.

B. Symbolic engine

SAPEC (Symbolic Analysis Program for Electric Circuit) calculates the symbolic Laplace domain network functions of a given circuit in expanded format with a variable "symbolism level": each component can appear in the network function with its name (a "symbol") or with its numerical value. The simulation module works on a netlist driven by Spice-like rules, which is automatically generated by the schematic editor. This netlist can also be edited or modified as an ordinary text file.

Right now, the program implements, for a symbolic analysis of a circuit, a modified version of Laplace determinant expansion. The algorithm drastically improves the classical complete tableau method, especially for sparse matrix cases. However, the algorithm is still sub-optimal with respect to the two-graphs methods, which is scheduled to be used in future program releases. An exhaustive survey and dissertation on symbolic analysis methods is given in [\[7\]](#).

An approximation routine is included in the program. When used, this routine reduces the symbolic expression to the most significant terms and suppresses the others. The terms to be suppressed are selected by the algorithm on the basis of a given maximum allowable error imposed by the user. This feature is particularly useful in the analysis of medium and large size circuits, where a full symbolic analysis results in long and cumbersome expressions. Actually, two types of approximation are possible: a) each coefficient of s is approximated by means of the elimination of the least significant terms. This is done if its nominal value is smaller than a given magnitude; b) the least significant terms of each coefficient of s are replaced by their numerical values and only the dominant terms keeps in a symbolic form in the final network function. In the approximation routine, two different techniques are integrated into the same algorithm: a "local" and a "global" technique [\[8\]](#). The local approximation gradually removes all terms of the final expression which have an absolute value smaller than a given magnitude, depending on the user supplied error value and the original expression mean value. A fine tune of the effective error is then obtained by applying a global approximation on the remaining part of the expression. This approximation is stopped when the calculated error is lower than the error previously set by the user.

C. Graphic postprocessor

WINGRF (the graphical postprocessor) shows the network function and displays the gain and phase plots, time delay, pole and zero position diagrams, time domain voltage and current waveform of step and pulse response. WINGRF gives the user the chance to activate a parametric option, which gives different curves of the same electrical quantity related to different values of a components selected as "parametric". Because the SAPWIN simulation engine only gives output in Laplace form, in the time domain the impulse response and step response are obtained by numerical Laplace transform inversion, using the Pade' approximation and stepping algorithm [9], [10]. If the function is not proper, the algorithm is preceded by long-divisions performed by the Vlach algorithm [11]. For pole and zero computation, Moore's method [12] is used.

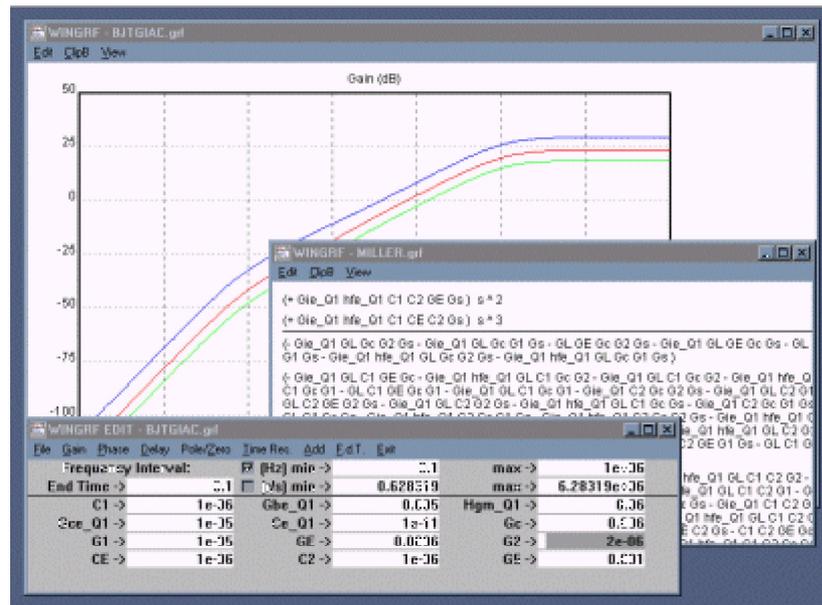


Fig. 3. Sapwin Ver.1.2 graphic processor.

III. An application example

There are many applications where symbolic tools are really useful. Some clear examples about the use of symbolic analysis tools in traditional courses of circuit theory can be found in [1]. Note that a professional (or "industrial") use of an electrical symbolic simulator should be distinguished from an educational one.

A simple example among the many possible given could be taking into consideration the LC-ladder elliptic low-pass filter shown in Fig. 4. This figure also shows the symbolic output. In the program schematic environment, the student can trace a possible filter implementation and assign the component values.

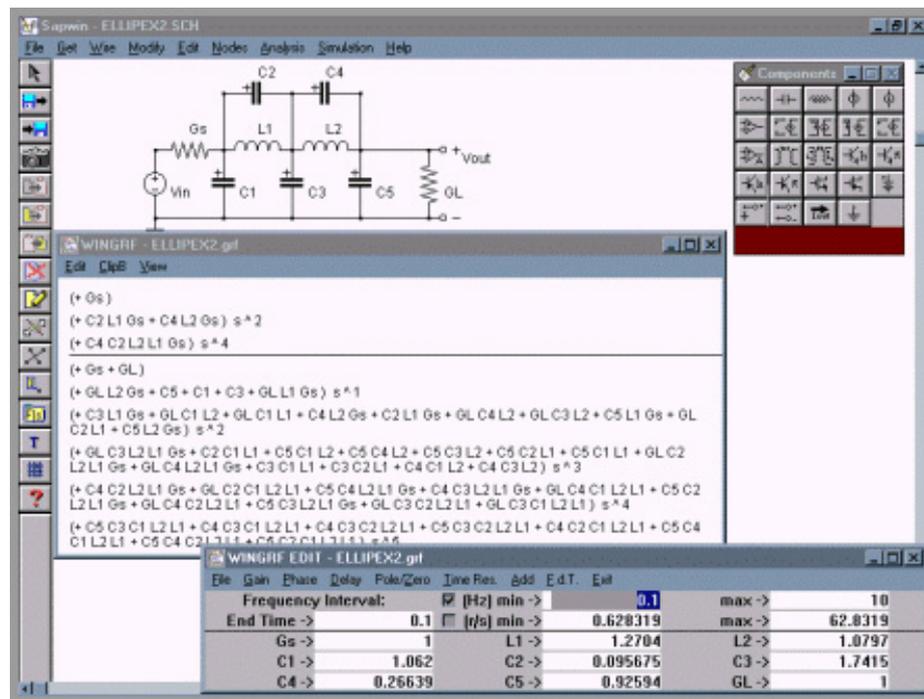


Fig. 4. LC-ladder elliptic filter - symbolic transfer function and component values.

As shown in by Fig. 4, a student can easily draw the circuit, generate the netlist for the symbolic simulator (not shown in figure) and obtain the Laplace domain transfer function. Moreover, the schematic entry and the symbolic transfer function windows let the student observe that this circuit, with seven reactive components and three capacitor loops (C_1 - C_2 - C_3), (C_3 - C_4 - C_5) and (C_1 - C_2 - C_4 - C_5), generates a fifth-order transfer function (a maximum degree 5 for s is shown in Fig. 4). Then, only two capacitor loops effectively decrease the degree of the network because two are loops of this kind, which include all the capacitors (there are no new capacitors, the third loop does not contain any capacitor which is not already included in the previous loops). This simple and straightforward representation is one of the main features of the [SAPWIN](#) symbolic simulator.

In basic electrical engineering courses, networks with loop constraints are explained and advanced filter-design courses deal with this topic more accurately.

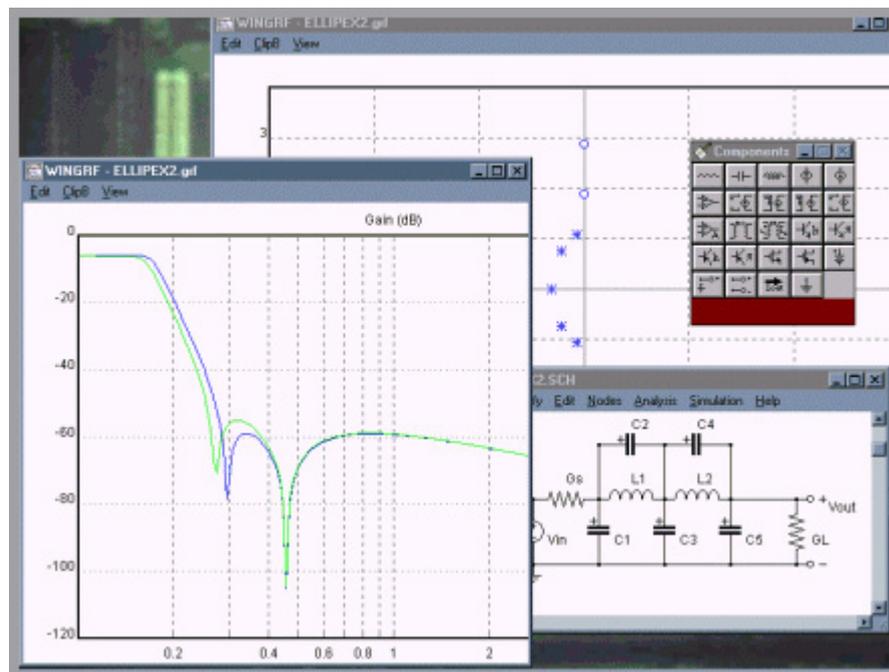


Fig. 5. LC-ladder elliptic filter - gain response, poles and zeros.

Fig. 5 shows the LC low-pass elliptic filter of Fig. 4, along with the gain response and position of poles and zeros on the complex plane. By using the symbolic expression of the filter transfer function, the student can control the components of the circuit most affecting the circuit behavior (e.g., DC gain, pole position and therefore band frequencies, etc.). The values of one or more components can be modified and the effect of this changing of values is shown by the program, with no need to run the circuit analysis again every time a component value is changed. This allows for a straightforward sensitivity evaluation and an immediate determination of dominant terms, also performing an approximation of the set up percent on the final expression.

IV. Recent developments and Sapwin 2.0

The program herein described and distributed with this issue is version 1.2 of [SAPWIN](#). The improvements scheduled for this program are resulting in the new version **SAPWIN 2.0**. This is still a beta version that will be tested and then delivered to those people who will consider it a useful tool in the engineering educational field. In this release, the graphical interface is going to be made more "user friendly," as shown in Fig. 6. SAPWIN 2.0 is a completely MDI (Multiple Document Interface) environment, where users can draw, load and save schemes of more circuits at the same time. Schematic editors, simulator engine, and graphic processors will be integrated into the same environment, avoiding the proliferation of multiple instances of the same executable.

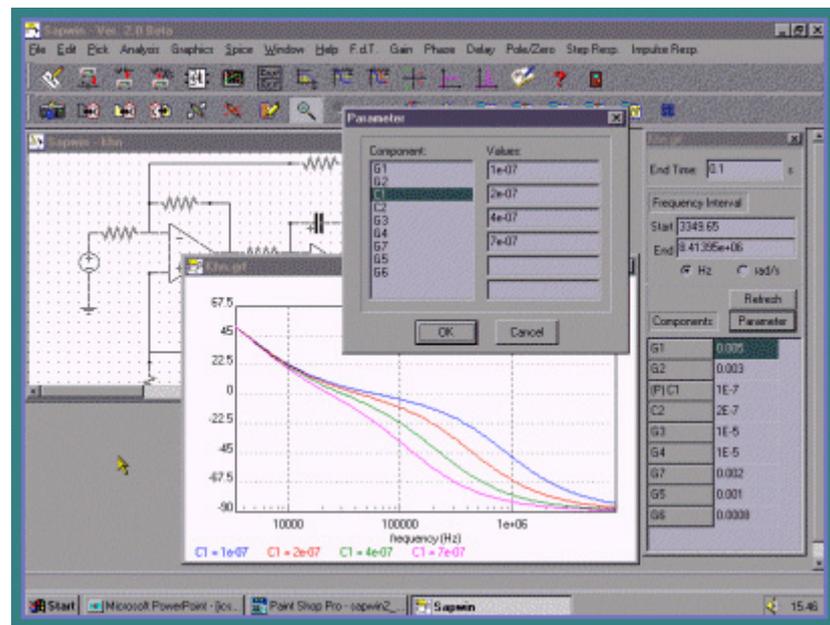


Fig. 6. Sapwin Ver.2.0 Beta screen.

A new program feature will be the possibility to add self-defined blocks to the circuit. This will allow the user to indicate the model of the device to insert in the overall scheme as a small sub-circuit. Each model may be characterized by a certain number of elements and inserted into an indexed library. This provides both the teacher and the student with good flexibility in designing and testing the circuit where the model of some device is assigned. **SAPWIN 2.0** will continue to give to students and teachers the possibility to work with a numerical simulator, in order to provide the student with an easy and fast method to compare two different and complementary methods to analyze and/or design analog electronic circuits by using only one platform.

Note that the program package is completely developed in a C++ object-oriented language, which is highly modular. This results in good flexibility and code re-usability. The program is used "as is" during basic electric science or network theory courses, but it is a flexible platform to develop further applications useful in advanced classes and research activity for people able to work on the program source code, modifying and training it to reach their objectives.

References

- [1] L.P. Huelsman, "Symbolic Analysis – A Tool for Teaching Undergraduate Circuit Theory." *IEEE Transaction on Education*, Vol. 39, N° 2, pp. 243-250, May 1996.
- [2] H.R. Pota, "Computer-Aided Analog Electronics Teaching", *IEEE Transaction on Education*, Vol. 40, N° 1, pp. 22-35, Feb. 1997.
- [3] A. Liberatore, A. Luchetta, S. Manetti, A. Reatti, "Frequency Domain Analysis of DC/DC Converters using a Symbolic Approach", *Proc. 1995 IEEE Int. Symposium on Circuits and Systems (ISCAS'95)*, Seattle, WA, USA, May 1995.
- [4] G. Fedi, R. Giomi, A. Luchetta, S. Manetti and M.C. Piccirilli, "On the Application of Symbolic Techniques to the Multiple Fault Location in Low Testability Analog Circuits", *IEEE Transaction on Circuits and Systems II*, vol. 45, pp.1383-1388, Oct. 1998.
- [5] A. Liberatore, A. Luchetta, S. Manetti and M.C. Piccirilli, "A new symbolic program package for

the interactive design of analog circuits”, in *Proc. of International Symposium on Circuits and Systems (ISCAS'95)*, Seattle, WA, USA, May, 1995, pp. 2209-2212.

[6] L.P. Huelsman, “SAPWIN, Symbolic Analysis Program for Windows - PC Programs for Engineers”, *IEEE Circuits and Devices Magazine*, Vol. 6, N° 2, pp. 4-6, Mar. 1996.

[7] P. Wambacq, G. Gielen and W. Sansen, “Symbolic Network Analysis Methods for Practical Analog Integrated Circuits: A Survey”, *IEEE Transactions on Circuits and Systems II*, vol. 45, n° 10, pp.1331-1341, Oct. 1998.

[8] G. Gielen, W. Sansen, *Symbolic analysis for automated design of analog integrated circuits*, Kluwer Academic Publishers, 1991.

[9] K. Singhal, J. Vlach, “Computation of time domain response by numerical inversion of the Laplace transform”, *Journal of the Franklin Institute*, Vol. 299, N° 2, pp. 109-126, Feb. 1975.

[10] K. Singhal, J. Vlach, “Method for computing time response of systems described by transfer functions”, *Journal of the Franklin Institute*, Vol. 311, pp. 123-130, 1981.

[11] J. Vlach, *Computerized approximation and synthesis of linear network*, Wiley, New York, 1969.

[12] J. B. Moore, “A convergent algorithm for solving polynomial equations”, *ACM Journal*, pp. 311-315, April 1967.

Author Contact Information

Antonio Luchetta

DIFA - University of Basilicata <http://www.difa.unibas.it/>

Contrada Macchia Romana Potenza, Italy

Phone: +39 0971 205150

Fax: +39 0971 205160

E-mail: luchetta@unibas.it

<http://www.difa.unibas.it/users/aluchetta/>

Stefano Manetti

DET - University of Florence <http://www.die.unifi.it/>

Via S. Marta, 3 Firenze, Italy

Phone: +39 055 4796282

Fax: +39 055 494569

E-mail: manetti@ingfi1.ing.unifi.it

Alberto Reatti

DET - University of Florence <http://www.die.unifi.it/>

Via S. Marta, 3 Firenze, Italy

Phone: +39 055 4796565

Fax: +39 055 494569

E-mail: mailto:reatti@ingfi9.die.unifi.it

Author Biographies

Antonio Luchetta received the degree in Electronics Engineering from University of Florence, Italy, in 1993. He is a Researcher in the Department of Engineering and Environmental Physics at the University of Basilicata, Italy, where he also teaches electrical sciences. His research interests include software and algorithms for electric network simulation, neural network application to forecasting and control problems, and elaboration of meteorological data. He is member of IEEE and AEI.

Stefano Manetti received the degree in electronic engineering from the University of Florence, Italy, in 1977. From 1977 to 1979 he had a research fellowship at the Engineering Faculty of Florence University. He was assistant professor of Applied Electronics at the Accademia Navale of Livorno from 1980 to 1983, and researcher at the Electronic Engineering Department of Florence University from 1983 to 1987. Since May 1987, he has been associate professor of Network Theory at the same University. In 1994, he join the University of Basilicata, in Potenza, Italy, as a full professor of Electrical Sciences. Since November 1996, he is full professor of Electrical Sciences at the University of Florence. His research interests are in the fields of circuit theory, neural networks and fault diagnosis of electronic circuits. He is member of IEEE, ECS and AEI.

Alberto Reatti received the degree in Electronics Engineering in 1988 and the Ph.D. degree in 1993, both from the University of Florence, Italy. Since 1992, he has been Associate Researcher with the department of Electric Engineering, Wright State University, Dayton, OH. In 1994, he was Visiting Professor with the Department of Electronics at the University of Siena. He was also Researcher at the Department of Electronics and Telecommunications at the University of Florence where is currently working as an Associate Professor. His interests are in the electric network simulation and high-frequency high-efficiency power converters.

[Return to Table of Contents](#)