

Validation of Symbolic Expressions in Circuit Analysis E-Learning.

Ludo Weyten, Pieter Rombouts, Benoit Catteau, and Maarten De Bock

This document is an authors draft version submitted for publication to IEEE Transactions on Education.

The actual version was published as:

Ludo Weyten, Pieter Rombouts, Benoit Catteau en Maarten De Bock 'Validation of Symbolic Expressions in Circuit Analysis E-Learning,' IEEE Transactions on Education, Vol. 54, No. 4, pp. 564-568, Nov. 2011.

Validation of Symbolic Expressions in Circuit Analysis E-Learning.

Ludo Weyten, *Member, IEEE*, Pieter Rombouts, *Member, IEEE*,
Benoit Catteau, *Member, IEEE*, and Maarten De Bock *Member, IEEE*

Abstract—Symbolic circuit analysis is a cornerstone of electrical engineering education. Solving a suitable set of selected problems is essential to developing professional skills in the field. A new method is presented for automatic validation of circuit equations representing a student's intermediate steps in the solving process. Providing this immediate feedback may strongly enhance the training effects. The new method was embedded in a web-based e-learning system and has proved to be useful in circuit analysis training, both at an introductory level and for more advanced problems in analog electronics.

I. INTRODUCTION

Symbolic analysis of linear networks is the backbone of the design of electronic circuits and systems [1]. Therefore a large body of literature is available on automatic symbolic analysis methods and their performance. The symbolic approach is also considered to be most valuable in teaching analog electronics [2], [3]. In both cases, the basic task consists of obtaining an algebraic expression for some specified network function.

In the teaching context, however, another type of problem may emerge with the ongoing automation in e-learning. The teaching of circuit analysis is initiated by explaining fundamental principles, introducing simple element models and studying some basic circuits. Then the student's skills are fully developed by exposing her/him to a suitable set of fresh problems [4], [5]. At this training stage, fast feedback by the instructor is an essential element of success. Providing feedback on an individual basis often turns out to be repetitive, hence any degree of automation might well be welcomed. So being able to make an automatic validation of a student's intermediate problem-solving steps may become an important issue with the emergence of more sophisticated learning systems for circuit analysis courses.

L. Weyten, P. Rombouts, B. Catteau and M. De Bock are with the dept. ELIS at the Engineering Faculty of the University of Ghent, Belgium.

Advanced systems for course delivery have been developed at several universities [4]-[8], providing homework pages with on-line automatic grading. These systems generate homework assignments with different sets of numerical values [7], [8] and automatically check the student's numerical results.

The focus in this paper, however, is on symbolic methods. The formulation of this task differs from the classic one addressed in symbolic network analysis: here an algebraic equation is given and the question is whether or not it holds true. Of course, one might compare the students' input to a set of previously-derived expressions stored in a database. A web-based trainer using this approach was presented in [9]. Usually, for non-trivial problems, several different solution paths may exist. Obviously, an appropriate normalization of input and reference data is also necessary. Since students tend to be very creative in discovering unusual solutions paths, gathering an adequate set of equations could be time consuming.

Another approach might be to use an (online) symbolic analysis tool to calculate a suitable network function for the given circuit, and to compare this to the student's answer. One might also substitute suitable sets of numeric values in the student's symbolic formula and use a standard numerical simulation tool. In both cases it is not always easy to find out which network function to derive for a given student's statement. Moreover, the student may previously define some symbols the symbolic network analysis tool is not aware of (e.g. an equivalent impedance for part of the circuit).

In this paper, a simple method is presented to verify the students' symbolic equations directly, using the circuit's schematic. It is first introduced for the large class of planar networks without controlled sources and operational amplifiers. It is then later shown how this can be extended to cover all linear circuits.

II. THE NEW VALIDATION METHOD

In general, the behavior of a network is governed by three sets of equations: Kirchhoff's voltage laws, Kirchhoff's current laws, and the set of current/voltage relations valid for the individual branches.

The two most common approaches to solving traditional linear network problems are based upon the introduction of a set of new variables (symbols). Either a set of node voltages or a suitable set of loop currents can be used. In the first case, Kirchhoff's voltage laws are automatically fulfilled. Solving the problem proceeds by writing down Kirchhoff's current laws combined with the branches' current/voltage relations. The second approach is the dual of the first one: by introducing appropriate loop currents, Kirchhoff's current laws are automatically fulfilled. The problem is solved by combining Kirchhoff's voltage laws with the branches' current/voltage relations.

Both sets of entities (basic node voltages and loop currents) have an interesting property: the former defines a minimal set of control variables governing all voltages in the network, and the latter controls all currents in the network.

As mentioned above, the problem of validation is different from the traditional analysis problem. The new approach is based upon using both sets of control variables simultaneously, i.e. two sets of new symbols are introduced. For the voltages, a straightforward choice is to choose an arbitrary node as a reference and to allocate a new voltage symbol to each node. Then, any voltage difference in the network can be written symbolically as a difference of at most two of these voltage symbols. It may prove to be more elegant to introduce an external reference node in order to write any voltage difference in terms of exactly two basic voltage symbols.

The choice of a suitable set of independent basic current symbols is less obvious. For planar networks, the loop currents usually are associated with the circuit's meshes (loops containing no other loops). With each mesh, a (clockwise) "mesh current" is associated. The setup is completed by introducing a special loop encircling the whole circuit and associating an (anti-clockwise) "outer mesh current" with it. In doing so, each branch current can be written as the difference of exactly two mesh

currents.

Another well-known fundamental set of independent loop currents may be obtained by considering an arbitrary spanning tree and linking in each of the remaining edges (branches) of the circuit's graph. If each fundamental loop (current) is identified by a specific symbol, any branch current in the circuit can be written as a sum (or difference) of these current symbols. This choice is more general and not limited to planar networks, but the number of loop current symbols needed to represent an arbitrary branch current may be more than two.

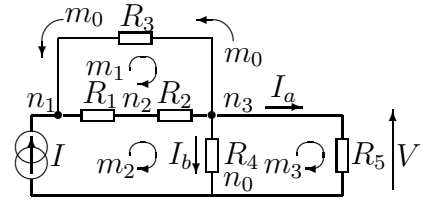


Fig. 1. A planar circuit example

In order to clarify the presentation of the new method, the simple planar circuit in Fig. 1 is used. The four nets in the netlist describing the connectivity of this circuit are associated with node voltages n_0, \dots, n_3 . Optionally, n_0 may be chosen as the reference node voltage symbol. Three (clockwise) mesh currents m_1, m_2, m_3 are used. The (anti-clockwise) outer mesh is identified by symbol m_0 . Now, each branch voltage can be written as the difference of two node voltage symbols and each branch current can be written as the difference of exactly two mesh current symbols.

Suppose the student's statement is

$$I = I_a + I_b. \quad (1)$$

Since only currents are involved, it is clear that only Kirchhoff's current laws are needed. Using the previously introduced mesh currents, the current symbols I, I_a and I_b in the student's statement are transformed as follows:

$$\begin{aligned} I &= m_2 - m_0 \\ I_a &= m_3 - m_0 \\ I_b &= m_2 - m_3. \end{aligned}$$

Thus, the student's statement (1) reduces to

$$m_2 - m_0 = (m_3 - m_0) + (m_2 - m_3).$$

Now the validation is straightforward: the symbol manipulator groups the coefficients per symbol:

$$(-1 + 1)m_0 + (+1 - 1)m_2 + (-1 + 1)m_3 = 0$$

and concludes that all terms are zero, indicating that the student's statement is true. This procedure is successful because Kirchhoff's current laws are implicitly satisfied due to the translations into fundamental mesh (current) symbols. Similarly, if the student's statement consists only of voltages, validation would be carried out using the basic node (voltage) symbol set.

But how should the validation of a more general statement, such as

$$V - \frac{R_4 R_5}{R_4 + R_5} I = 0 \quad (2)$$

be carried out? Here a two step approach is proposed: the first step is to observe that any resistance value symbol on the left side of (2) may be expressed in terms of a branch voltage divided by a branch current. (A pulsation symbol ω may also be needed in AC analysis). Then each branch voltage is transformed into the difference of the two corresponding fundamental node voltage symbols and each branch current is transformed into the difference of the fundamental mesh current symbols. Through these transformations, all relevant branch voltage/current relations are used. Kirchhoff's voltage laws as well as Kirchhoff's current laws are implicitly satisfied by introducing the new symbols. The resulting expression is in the independent fundamental symbols only (with coefficients ± 1).

For the example in Fig. 1 the transformations are

$$\begin{aligned} V &= n_3 - n_0 \\ R_4 &= (n_3 - n_0)/(m_2 - m_3) \\ R_5 &= (n_3 - n_0)/(m_3 - m_0) \\ I &= m_2 - m_0 \end{aligned}$$

and thus the left side of eq.1 simplifies to:

$$(n_3 - n_0) - \frac{(n_3 - n_0)(n_3 - n_0)(m_2 - m_0)}{(n_3 - n_0)(m_3 - m_0) + (n_3 - n_0)(m_2 - m_3)}$$

The algebraic symbol manipulator expands the denominator D into

$$D = (n_3 - n_0)[(n_3 - n_0)(m_3 - m_0) + (n_3 - n_0)(m_2 - m_3)] - (n_3 - n_0)(n_3 - n_0)(m_2 - m_0)$$

and finds an empty list of terms, indicating that statement (1) holds.

This new validation method was extensively tested in the web-based Circuit AnalySis Trainer CASTr described in Section VI. But before giving these results, some extensions and a special case are discussed in the next section.

III. EXTENDING THE METHOD TO CONTROLLED SOURCES AND NULLORS

As long as the controlled source is characterized by a symbolic parameter relating two currents or voltages, the approach described in the previous section remains valid. Since linear transistor or op amp behavior is usually described in terms of simple two-terminal elements and controlled sources, the method remains adequate here. A well-known circuit is shown in Fig 2. In a first example, the

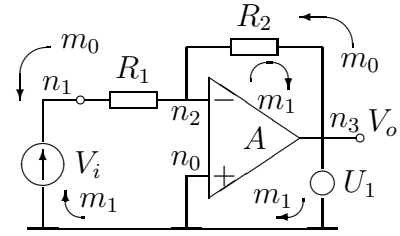


Fig. 2. Example with op amp

op amp is modeled as a voltage-controlled voltage source $U_1 = A(V_+ - V_-)$. The transformations for the element symbols R_1 , R_2 , A and the relevant voltages are:

$$\begin{aligned} V_i &= n_1 - n_0 \\ V_o &= n_3 - n_0 \\ R_1 &= (n_2 - n_1)/(m_0 - m_1) \\ R_2 &= (n_3 - n_2)/(m_0 - m_1) \\ A &= (n_3 - n_0)/(n_0 - n_2). \end{aligned}$$

Using these values, validating the result

$$V_o = \frac{-AR_2}{AR_1 + R_1 + R_2} V_i$$

becomes straightforward.

In a second example, the op amp is modeled as a nullor (infinite gain). The controlled source U_1 is considered as a norator, i.e. a device that can deliver an arbitrary current. The voltage between the op amp's input terminals is sensed by the nullator element controlling the norator, in order to make

$n_2 = n_0$. So the nullator is treated as a circuit element binding the two fundamental symbols n_0 , n_2 related to it. Of course, the equation transforming the symbol A becomes pointless. In this way, the result

$$V_o = -\frac{R_2}{R_1}V_i$$

is easily validated.

The supplementary relations associated with nullors are kept in a separate list. Since they always indicate two (similar type) fundamental symbols to be equivalent, this is simply a mapping from sets of symbols into specific representatives. This synonym table is used at the validation stage.

IV. THE SPECIAL CASE OF DEAD SUB-CIRCUITS

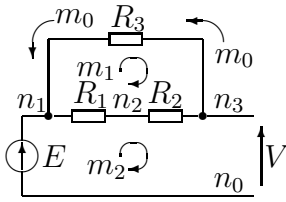


Fig. 3. Example with “dead” sub-circuit

Sometimes a sub-circuit may be linked to the embedding circuit through a single node (net) only. An example is shown in Fig. 3 (sub-circuit R_1, R_2, R_3). If no independent sources are present in this sub-circuit, and furthermore no dependent sources are driven from outside the sub-circuit, all of its branch currents are zero and all of its node voltages are equal, regardless of the circuit’s element values.

In the e-learning system CASTr, described in Section VI, such “dead” sub-circuits are automatically detected and the corresponding set of node voltages is registered in the synonyms list. The pairs of meshes corresponding to branch elements present in the sub-circuit are treated similarly.

V. OTHER EXTENSIONS

A. Non-Planar Networks

For non-planar networks, any set of suitable fundamental loop currents could be used. But it is also possible to handle this case by the procedure meant for planar networks, through the introduction of a new circuit element designated as a “bridge” (wires crossing). This element links two pairs of

mesh current symbols to two other such pairs. This information may be kept in a list of supplementary relations.

B. Feedback Factor and Eigenvalues

The approach can also be used to validate other algebraic relations derived from the given circuit. A student’s expression giving the feedback factor or the loop gain around an operational amplifier may also be checked.

First, all independent sources are set to zero (e.g. short-circuited or eliminated). Then the pair of node voltages corresponding to the controlling (input) voltage is considered, as are the pair associated with the controlled quantity (voltage) in the Thevenin’s equivalent at the op amp’s output. Two different relations exist between those node voltage pairs. In the first place, the two signals are linked by the amplifier gain factor. The second relation is simply the feedback factor. So in order to validate the student’s expression for the feedback factor, all its symbols are substituted by their equivalent in terms of fundamental node voltages and mesh currents (as per the usual procedure). Regrouping the terms using the algebraic symbol manipulator leads to the final conclusion as to the correctness of the student’s expression.

The validation method can also be used to check the (Laplace domain) equations determining the circuit’s eigenvalues. Suppose the student claims

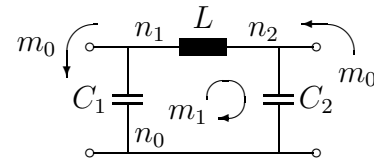


Fig. 4. Circuit example for eigenvalue problem

that the poles of the circuit in Fig. 4 are given by

$$s^2 L \frac{C_1 C_2}{C_1 + C_2} + 1 = 0. \quad (3)$$

The transformations for the element symbols in Fig. 4 are derived from the schematic:

$$\begin{aligned} C_1 &= (m_0 - m_1)/s(n_1 - n_0) \\ C_2 &= (m_1 - m_0)/s(n_2 - n_0) \\ L &= (n_1 - n_2)/s(m_1 - m_0). \end{aligned}$$

Thus the left of eq.3 simplifies to:

$$s^2 \frac{(n_1 - n_2)}{s(m_1 - m_0)} \cdot \frac{\frac{(m_0 - m_1)}{s(n_1 - n_0)} \cdot \frac{(m_1 - m_0)}{s(n_2 - n_0)}}{\frac{(m_0 - m_1)}{s(n_1 - n_0)} + \frac{(m_1 - m_0)}{s(n_2 - n_0)}} + 1.$$

Expanding this expression leads to an empty list of terms, and it may be concluded that the student's equation (3) holds.

Since this approach is based just upon linearity and Kirchhoff relations, it may be useful for other applications in electrical engineering, for example in charge-centered switched capacitor circuits.

VI. IMPLEMENTATION IN AN E-LEARNING SYSTEM

Along the lines suggested in Section I, a new Circuit AnalySis Trainer (CASTr) was developed. This is a php- and MySQL-driven web-application.

The basic idea of collecting the elementary steps of the student's reasoning through a web-based interface and providing immediate feedback was presented in a previous paper [9], but the new automatic validation method considerably enhances its performance and usability.

Initially, the web interface shows only the problem question and the accompanying schematic. The student is expected to tackle the problem step by step through a sequence of statements. Each acceptable step taken by the student is summarized permanently on the screen as a single line, Fig. 5.

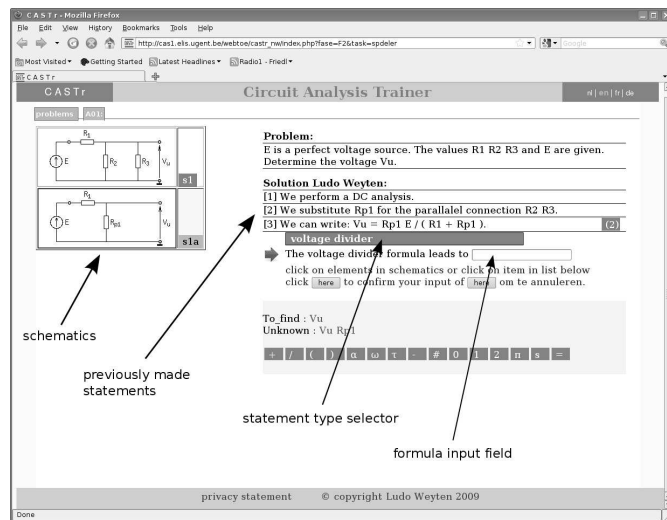


Fig. 5. Screen dump of the web-interface (English version)

Steps involving the definition of valid new current or voltage symbols are also echoed in the schematics. For steps involving circuit transformations (e.g.

introducing the equivalent impedance of a series connection), the resulting new schematic is generated automatically by the system. The third category of statements are equations assembled by the student, either by using the keyboard or by selecting (clickable) symbols from the circuit's schematic.

In [9] each new input from the student was validated (after normalizing) to a set of possible correct statements that had been previously stored in a database table. A serious drawback of this approach is the need for the instructor to populate the database with all the correct statements for each problem. Of course, several possible solution paths must be provided initially. In the authors' experience, valid solutions may easily be overlooked.

Using the new method described in Section II, the correctness of each equation generated by a student can be verified automatically, using merely the circuit's connectivity as derived directly from the schematic. This alleviates the tutor's work to a large extent: a reference set of correct statements is no longer needed and even steps along very uncommon solution paths are processed correctly. The tutor only has to specify acceptable models, assumptions (e.g. nullator, no saturation) and methods for solving the problem.

As soon as the circuit's schematic is loaded into the system, a set of fundamental node voltages for each branch is obtained through a simple net-list extraction algorithm: each net number is the index of the corresponding independent voltage symbol. Then a set of independent meshes is determined. For each element symbol the two net indices and the two mesh indices are stored in the net list table.

Each element also has a pointer to its type. For each element type (R,C,L,...) a specific recipe is stored describing the voltage/current relationship involved. For impedances, the symbols s and ω are also needed for representations in the Laplace or frequency domain.

The system is now ready to accept statements from the student. Each symbol in the input string is looked up in the net-list table and it is replaced by the corresponding expression in terms of base voltage and current symbols (identified by their indices). The student input is then expanded by an algebraic manipulator routine and the correctness of the result is checked. This approach is adequate in nearly all situations. Exceptions are handled as mentioned in Section IV.

Determining whether the student's statement set is sufficient to solve the problem at hand requires a separate processing step. Here the presence of more than one unknown quantity in the student's expressions indicates that further input is needed.

VII. EVALUATION

The new interactive circuit analysis trainer was tested during the academic years 2008-2009 and 2009-2010 with third-year Bachelor's students in Electrical Engineering at Ghent University, in an Analog Electronics course. It was used on a voluntary basis by 76 students.

A. Reliability of the Method

Each student's activity was monitored and appropriate reports were generated from the system's database. In 2008-2009 a set of 30 exercises was used. This was extended to 50 in 2009-2010. Difficulty levels ranged from very basic problems to more complex circuits using operational amplifiers and even simple transistor models.

Each combination of a student and an initiated exercise was considered: over the two academic years, 1730 such combinations were observed. On the average, the students initiated some 23 problems (30 in 2009-2010). The students solved 1657 out of the 1730 initiated problems (i.e. 96%).

The system's automatic validation routines processed 6374 equations delivered by the students. Whenever unexpected behavior was reported by a user, the validation process (program code) was thoroughly verified. In the initial phase, some bugs in the program were discovered and fixed by this approach, but after some time no more irregularities were seen. Hence the validation method turned out to be reliable.

B. Assessment by Students

After completing the course, the students were asked to provide feedback on their experience with the system, both in 2008-2009 and in 2009-2010. In order to encourage an unreserved response, the survey was organized anonymously. A total of 40 participants filled out the questionnaire. The results are presented in Table I.

The questionnaire was set up with answers on a five-point Likert-like scale (strongly disagree /

disagree / neutral / agree / strongly agree). The respondent could also formulate any useful (free text) remarks. Most respondents considered the sys-

| statement about the application | disagree | | neutral | agree | |
|---------------------------------|----------|----|---------|-------|----|
| | -- | - | 0 | + | ++ |
| I used it several times | 1 | 1 | 4 | 15 | 19 |
| is a useful tool | 2 | 1 | 11 | 18 | 8 |
| is user-friendly | 2 | 3 | 20 | 13 | 1 |
| interface is good-looking | 1 | 4 | 12 | 19 | 3 |
| direct feedback is useful | 1 | 3 | 7 | 22 | 6 |
| replacement for homework | 0 | 11 | 10 | 13 | 6 |
| complement to homework | 0 | 1 | 6 | 22 | 11 |

TABLE I
ASSESSMENT BY STUDENTS IN 2008-2009 AND 2009-2010

tem to be useful. They agreed that it was a nice complement to traditional homework. The students generally appreciated the user interface. Nevertheless, several user interface improvement possibilities were distilled from the respondents' free remarks.

VIII. CONCLUSION

In this paper a new web-based system for training students in the framework of teaching electric circuit theory and electronics was presented. The reader may visit the demo-website [10] for a more elaborate try-out of the system. The novelty of the approach is that it is centered on symbolic analysis techniques (essential in a design context) and that any of the student's equations are validated immediately using data automatically extracted from the circuit's schematic. Furthermore, it tracks the student through all the steps of his or her reasoning path, rather than just accepting or rejecting the final outcome.

The student's reasoning is tracked very closely (all steps are logged, including wrong ones). The frequent occurrence of specific error types may indicate less-understood topics (either for individual students or for the whole group). So the system can be used to obtain valuable information for the tutor about possible improvements in his teaching.

REFERENCES

- [1] L. Huelsman, "Symbolic analysis – a tool for teaching undergraduate circuit theory," *IEEE Transactions on Education*, vol. 39, no. 2, pp. 243–250, May 1996.
- [2] H. Pota, "Computer-aided analog electronics teaching," *IEEE Transactions on Education*, vol. 40, no. 1, pp. 22–35, February 1997.

- [3] J. Teng, J. Fidler, and Y. Sun, "Symbolic circuit analysis using mathematica," *International Journal of Electrical Engineering Education*, vol. 31, no. 4, pp. 324–332, 1994.
- [4] A. Luchetta, S. Manetti, and A. Reatti, "Sapwin - a symbolic simulator as a support in electrical engineering education," *IEEE Transactions on Education*, vol. 44, no. 2, May 2001.
- [5] R. Assaad and J. Silva-Martinez, "A graphical approach to teaching amplifier design at the undergraduate level", *IEEE Transactions on Education*, Vol. 52, no. 1, pp. 39–45 February 2009.
- [6] B. P. Butz, M. Duarte, and S. M. Miller, "An intelligent tutoring system for circuit analysis," *IEEE Transactions on Education*, vol. 49, no. 2, pp. 216–223, May 2006.
- [7] L. Palma, R. F. Morisson, P. N. Enjeti, and J. W. Howze, "Use of web-based materials to teach electric circuit theory," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 729–734, November 2005.
- [8] C. R. Smaill, "The implementation and evaluation of oasis: A web-based learning and assessment tool for large classes," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 658–663, Nov. 2005.
- [9] L. Weyten, P. Rombouts, and J. De Maeyer, "Web-based trainer for electrical circuit analysis," *IEEE Transactions on Education*, vol. 52, no. 1, pp. 185–189, Feb. 2009.
- [10] L. Weyten. (2010, Oct 1). CASTr [Online]. Available: <http://castr.be> and <http://cas1.elis.ugent.be/castr>



Benoit Catteau Benoit Catteau was born in Roeselare, Belgium in 1981. He obtained the Ir. degree in electronics from Ghent University, Belgium in 2004. After graduating, he joined the Department of Electronics and Information, where he obtained the Dr. degree in 2010. His technical interests are signal processing and circuits and systems theory.



Maarten De Bock Maarten De Bock was born in Sint-Niklaas, Belgium in 1985. He obtained the Ir. degree in electronics from Ghent University, Belgium in 2008. In the same year he joined the Electronics and Information Systems Department of Ghent University as a Research and Teaching Assistant at the Circuits and Systems group. His technical interests are signal processing and analog circuit design.



Ludo Weyten Ludo Weyten was born in Mortsel (Antwerpen, Belgium) in 1947. He received the Ir. degree and the Dr. degree from Ghent University (Belgium) both in Electrical Engineering, in 1970 and 1978 respectively. From 1970 to 1972 he joined the Electronics Laboratory of the same university as a Research Assistant. From 1972 to 1975 he was teaching at the National University of Zaïre (now Congo) at Lubumbashi in a government technical co-operation project. Since 1975 he is with the Engineering Faculty of Ghent University (Belgium) where he is currently Professor. His teaching and research interests are in the field of electronics circuits and systems and e-learning.



Pieter Rombouts Pieter Rombouts was born in Leuven, Belgium in 1971. He obtained the Ir. degree in applied physics and the Dr. degree in electronics from Ghent University in 1994 and 2000 respectively. Since 1994 he has been with the Electronics and Information Systems Department of the Gent university where he is currently Professor. In 2002 he was a visiting professor at the University Carlos III in Madrid (Spain). His technical interests are signal processing, circuits and systems theory and analog circuit design.