90-515

Experiences with a Personal Computer Network in Electrical Engineering Education

JOOS VANDEWALLE, SENIOR MEMBER, IEEE, DIRK RENIERS, AND BART DE MOOR, MEMBER, IEEE

Abstract—At the ESAT laboratory (Electronics, Systems, Automation, Technology) of the K.U. Leuven in Belgium, extensive efforts have been made on the implementation of exercise sessions on a PC network. Since 1986, 13 different exercise sessions of eight different courses take place on the network for 300 students of the last three years of electrical engineering. The courses cover several aspects of the electrical engineering education: linear and nonlinear systems, switching and control theory, electronic measurements, digital data transmission, etc. All the exercise sessions are menu driven and have the same pattern so that the students do not need a background on computers or software. The use of a computer permits the student to spend more time on testing and analyzing several solution methods, conditions, restrictions and properties of certain techniques, etc., instead of wasting time on boring repetitive calculations. First of all, an example that demonstrates the advantages of using a personal computer to solve an exercise is given, followed by the purposes of the network. The configuration of the network, its hardware, and software are presented. The contents of each session is briefly discussed and one session is explained more in detail. Student evaluations are described that indicate the success of the exercise sessions on the network.

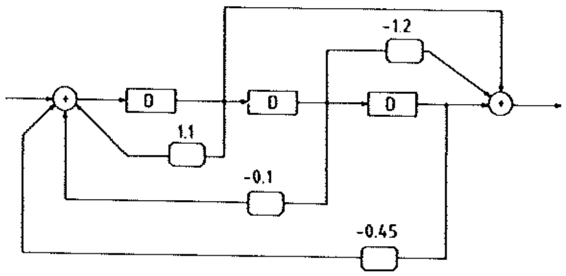

Fig. 1. A typical exercise of the course signals and systems. (a) For the given system, calculate the state equations and the difference equation. (b) Calculate the impulse response numerically by reasoning on the system. Calculate the general analytic expression, starting from the difference equation. (c) Calculate the transfer function and verify it with the impulse response (use your tables). Calculate the poles and zeros. Is the system input–output stable? (d) Calculate the response for:

$$u[k] = \text{Cos}\,(2k)$$

$$u[k] = \text{Cos}\,(2k + 1) + 2\ \text{Step}\,(k).$$

I. Motivation

FOR an engineering student, it is necessary to take exercises in order to assimilate several concepts, relations, design methods, and engineering techniques. This, however, can be very time consuming. Let's take a typical sample problem of the course on signals and systems of the third year of electrical engineering (see Fig. 1). It takes a student 30 min to solve this problem. The same student can read the exercise, model it, enter it in a PC and solve it in 14 min. This means that half of the time is left for experimentation with questions of the following nature: what would happen if some poles were omitted or if that parameter changes, what would happen to the impulse response if I shift the complex poles, etc.

II. Engineering Education in Belgium

In Belgium, the engineering education consists of two parts: the candidatures and the technical years. During their first two years all engineering students have to take the same courses. They all have to learn several aspects of mathematics, the principles of physics, chemistry, mechanics, etc. After these candidatures, the students can choose between several specializations like electrical engineering, mechanical, computer, chemical, etc.

Almost all the courses of the engineering education consist of theory and exercises. The theoretical lessons are taught in the morning by professors to the entire class. The labs and exercise sessions are made in the afternoon in groups of 15 to 30 students under the guidance of an assistant and last between 2 and 2 1/2 h per session.

III. Goals of the Network

A personal computer can be very useful to engineering students while doing exercises on electrical topics:

- A PC can make fast calculations. This way the student can study problems that could not be solved with pen and paper because of their complexity. Instead of calculating, the student can do more creative work such as computing and studying several times the same problem, using different inputs or parameters in order to investigate the influence of changing conditions, the restriction and properties of certain techniques, etc.
- A PC has a lot of graphical possibilities: abstract principles like convolution, modulation, Bode plots, aliasing, bifurcation, root loci, transfer functions, limit cycles, and Nyquist curves can easily be visualized (see Fig. 2). Simulations of analog or digital systems at gate or transistor level can be done without any problem (see Fig. 3).
- A PC makes it possible for each student to work at his/her own speed and to repeat parts he/she did not understand quite well yet. The use of a PC gives the student more autonomy: he/she is forced to handle the problem

Manuscript received July 9, 1989.
The authors are with E.S.A.T. Lab., Department of Electrical Engineering, Katholieke Universiteit Leuven, 3001 Heverlee, Belgium.
IEEE Log Number 9038690.

0018-9359/90/1100-0314$01.00

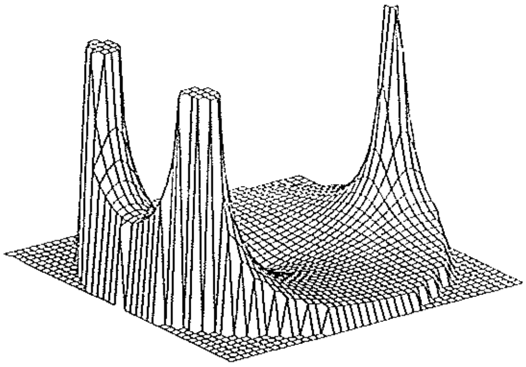

Fig. 2. A 3-D view of the absolute value of the transfer function of a system inside the unit circle of the complex plane taken from the session on time discrete systems using the program PC-Matlab (values larger than a certain number are cut off).

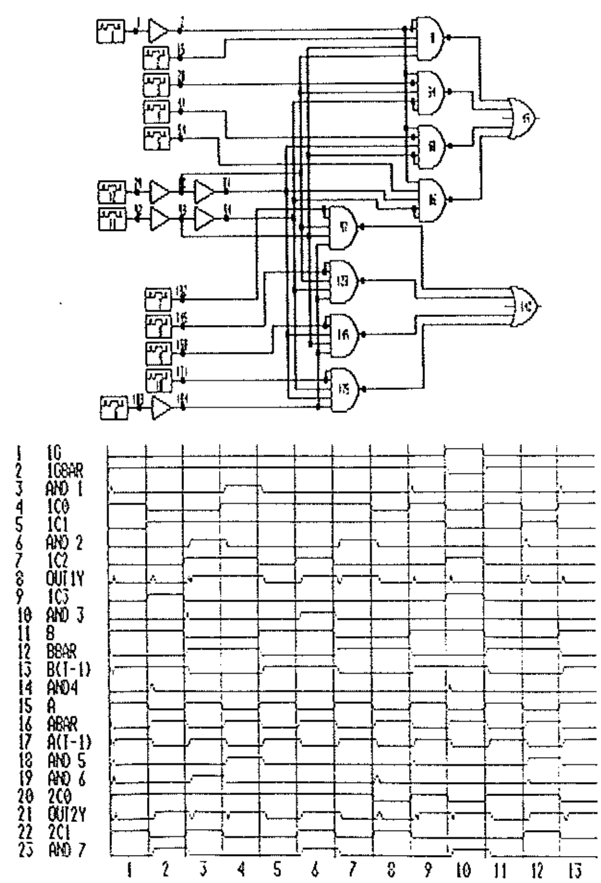

Fig. 3. A simulation of a digital circuit from the session of switching theory using the program Micro-Logic.

more actively than during a classic ex cathedra exercise session.

- By making exercises on the network the engineering student gets an opportunity to work on a personal computer which is always a useful experience.

IV. Configuration of the Network

In September 1986 ESAT started up a Local Area Network with 20 personal computers. PC's were chosen instead of other computers (e.g., a Vax-network or minicomputers) since the PC is nowadays a de facto standard in the industry since it is more flexible, easier to use, and represents a smaller investment.

The network (see Fig. 4) is an EtherLink network with three servers: a 3Com 3Server Model 3C1001 with a hard disk of 70 Mbyte, a PC-server Commodore PC 20 with a 20 Mbyte hard disk and a PC-server Commodore PC 40 with a hard disk of 2 × 20 Mbyte. The network operating system of the 3Server is 3Plus [1]. This is a powerful system with a lot of possibilities to manage the network. Unfortunately some of the software packages we use cannot run on 3Plus (when they are installed on a server with 3Plus, Micro-Logic [2] and PC-Matlab [3] have a lack of memory and Gem [4] cannot read all its files). That's why another server operating system was installed, EtherSeries [5], on the hard disk of a personal computer that became a PC-server. Since this system is rather slow when several users work on it, we installed EtherSeries twice so that the number of users per server is divided by two. A user can login on the 3server and on a PC-server simultaneously without any problem.

The several personal computers and the servers are linked with each other via coaxial cables; in each PC an EtherLink board Model 3C501 is installed.

V. Hardware

Initially 15 Commodores PC 10 with monochrome monitors and a AGA (Advanced Graphics Adapter) graphical card were purchased. Beside the 2 PC-servers (Commodore PC 20 and 40) there's another Commodore PC 40 on the network with a colour monitor. There are also 2 IBM PC's, a Systech Plus 700 PC, and Olivetti M24, and a Sperry PC. Due to incompatibilities the Olivetti and the Sperry are not used in the network.

In each PC there is beside the processor (8088 or 8086) a 8087 coprocessor. Since a lot of programs use a MicroSoft Mouse, several mice are available. There is a Tecmar QIC 60 tape streamer to make back ups of the hard disks of the servers. There are two Brother printers linked on the network: a M1509 and a 2024L.

One PC can be linked with the Vax computer of our laboratories so files can be sent from the network to the Vax and vice versa.

VI. Software

There are a lot of software packages available on the network: scientific programs like Simnon, PC-Matlab, MS-Kermit, Phaser and Micro-Logic, graphical programs like Ventura, PC-Storyboard and the applications of Gem, databases and editors like Volkswriter, WordPerfect, MS-Word, MicroTex etc.

Whenever it was possible, software packages were used to make the exercises. Scientific programs are normally powerful, fast, thoroughly tested, and less expensive than self developed programs. Most exercise sessions are based on PC-Matlab, Phaser or Micro-Logic.

PC-Matlab is the PC version of Matlab (this stands for matrix laboratory). It is an interactive system with as basic data element a matrix that does not require dimensioning.

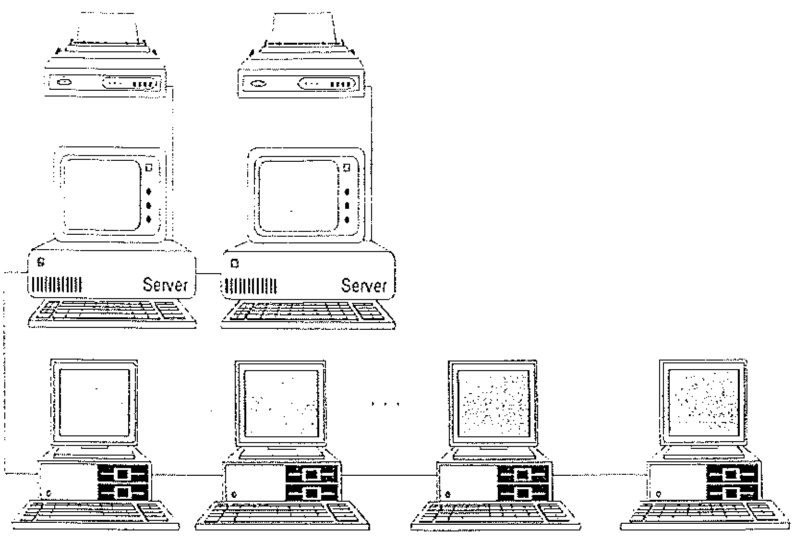

Fig. 4. The PC-network configuration with two servers, two printers, and 20 personal computers.

It allows the user to solve many numerical problems in a very short time. The program is very user friendly and it is very easy to insert commands. Graphical routines for two- and three-dimensional plots are available. Users can define their own command files and functions using loops (for . . . , while . . .), tests (if . . . then . . . else) and some basic routines for input and output of results (input, disp, plot etc.).

Phaser [6] is versatile, interactive, menu-driven program for the simulation of systems of nonlinear differential and difference equations. The user first creates a suitable window configuration for displaying a combination of views. Then he can specify various choices in preparation for numerical computations. The solutions can be manipulated graphically.

Micro-Logic II is an interactive, digital logic drawing, and simulation tool that's controlled by pull down menus. It allows the student to sketch (with a mouse) a logic drawing and automatically creates a netlist suitable for simulation. It provides a shape editor and contains a library of 200 components, including Boolean functions, data channels, clocks, flip-flops, and a number of common TTL/CMOS MSI logic functions.

VII. Overview of the PC-Sessions

All the exercise sessions run on the network are obligatory for the students of electrical engineering (two sessions for the students of the first technical year, six for the students of the second year, and five for the students of the last year).

When students come to the network to take an exercise session they just have to insert a special start floppy in drive *A* and to switch on the PC. They automatically log in and link with the servers. From a menu they can choose the exercise session they want to take. Experiments with the number of students that work on one PC showed clearly that two is the ideal number of users per PC. One user loses a lot of time and with three or more users there is at least one user who does not pay attention. So the sessions on the network are performed by two students per PC. There is always an assistant available for help concerning the program or the subject matter.

Here's a list of all the exercise sessions that take place on the network:

1) *Time Discrete Systems*—This session of the introductory course on signals and systems deals with linear time invariant, time-discrete systems. The student can define and simulate systems in order to get familiar with the different forms of representation of a system (block diagram, state equation, difference equation, transfer function) and with its properties like impulse response, eigenvalues, poles, etc. (See next section.)

2) *Signal Analysis*—This second session of signals and systems permits the student to define signals and manipulate these using filters, Fourier and inverse Fourier transformation, convolution, etc.

3) *Control Theory* [7]—The student defines linear systems and signals. He/she can ask a diagram of the location of the poles and zeros in the complex plane, examine an analytic approximation of a system by a second order system, time and frequency domain analysis, root loci, phase and gain margin, and the student can study Bode, Nyquist, and Nichols plots.

4) *Switching Theory* [8]—The student designs logic circuits at gate level and simulates these. The student has to study a multiplexer, a R-S flip-flop and a shift register and has to realize a J-K flip-flop, and a 4-b counter, given the state table.

5) *Information Transmission*—This program displays spectra of different signals and shows the user the effects of modulation (AM) and demodulation and of frequency modulation (FM) and demodulation.

6) *Phase Locked Loops* [9]—The first session of the course electronic measurements gives the definition of a linear phase-locked loop (PLL). The linear PLL is simulated (explanation of its transfer function, time and fre-

quency response) and some applications of a PLL are mentioned: amplitude modulation and demodulation and the controller.

7) *Digital Oscilloscope*—This program [10] on electronic measurements has been developed by the team of teaching assistants. It deals with the sampling of a signal and the influence of the sample frequency on the reconstruction of the original signal, with the pros and cons of a digital oscilloscope in comparison with an analog oscilloscope and with the signal processing features that enable the user to compare the original signal with the signal after processing, in the time domain or after a fast Fourier transform (FFT).

8) *Logic Analyzer*—In this session of electronic measurements the student learns the operation and the use of a logic analyzer [11]. A circuit with a counter and a ROM is used as a unit under test.

9) *Nonlinear Systems 1*—The student calculates and interprets graphically [6] nonlinear differential equations and nonlinear systems examining limit cycles, the presentation of trajectories and the simulation of harmonic oscillators, the three-body problem, and the jump phenomenon (see Fig. 5).

10) *Nonlinear Systems 2*—The student investigates the stability of nonlinear systems with the method of Popov, describing functions, linearization, and Nyquist, and examines the response of a nonlinear system on different initial states and inputs.

11) *Digital Electronic Systems*—The student designs and simulates logic circuits at the gate level. The student has to study a masterslave flip-flop and has to design two sequential circuits in pulse mode.

12) *Autonomous Shift Registers*—In the first session of digital data transmission [12] the students have to design and simulate a pseudorandom noise generator with ordinary components as flip-flops and exors.

13) *Algebraic Codes*—This program on digital data transmission enables the user to design and simulate an encoder–decoder that can restore a single bit error in a codeword by adding redundancy to this word.

VIII. An Elaborated Example: The Session of Discrete Time Systems

In order to illustrate the overall procedure the session of the course signals and systems of the first technical year about linear discrete time systems is described in detail. The content of this course is very close to the well known book of Gabel and Roberts [13]. The programs for this session (including the menu structure) are written using PC-Matlab commands. The main menu looks like this: 1) help, 2) demo, 3) exercises, 4) define systems and signals, 5) system characteristics, 6) response to an input or initial state, 0) stop, enter your choice (0 . . 6).

1) "Help" explains the user the purpose of the session and how the program can be used. There's a help function at each level of the menu. Sometimes it contains a summary of the subject matter.

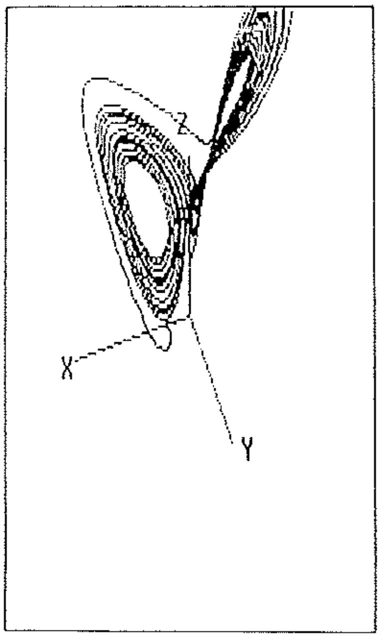

Fig. 5. A simulation of a three-body problem (circulation of a planet in the gravity field of two other planets) using the program phaser.

2) "Demo" gives a survey of all possible operations on the basis of a typical example.

3) "Exercises" gives a collection of problems which can be solved by the student; if necessary some data (like the definition of systems or signals) are loaded into the memory of the PC:

a) Investigate the stability of several systems: check which of the given systems are internally stable, input–output stable, marginally stable, or unstable.

b) Study first-order systems: see and explain the different impulse responses that can be obtained from a first-order system.

c) Study second order systems: for the given systems find the relation between the location of the two complex conjugate poles, the oscillation frequency, and the damping of the impulse response.

d) Realize systems with specified impulse responses: a step function, a shifted step function, a ramp function, a periodic function, and an undamped sine or cosine.

e) Analyze the frequency domain characteristics: examine the relation between the location of the poles of the given systems and the Fourier transform of their impulse responses.

f) Filtering interpretation of frequency characteristics: apply several sine input signals to a given system and compare the resulting gain and phase shift with the frequency characteristic.

g) Analyze the three-dimensional complex system representation: what is the relation between on the one hand the location of the poles of the given systems and on the other hand the three-dimensional plot (Fig. 2)? Same question for the relationship between the frequency characteristic and the three-dimensional plot?

h) Study autonomous systems: apply eigenvectors as initial state, look at the evolution of the state in the two-dimensional phase plane and explain the result.

i) Study the response of a system to input signals: use as input a pulse, a step, a polynomial, a sine, a damped sine, an exponential function or a combination of these or a periodic signal like a squarewave or a sawtooth and examine the output.

4) "Define systems and signals" enables the student to retrieve predetermined systems and signals, earlier loaded into the memory of his/her PC, or to create his/her own systems and signals which can be analyzed and simulated later on. The student can define a system in four ways: by its state space matrices, its difference equation, its transfer function, or its poles and zeros. The menu looks as follows:

Systems
1) Help
2) Matrices
3) Difference Equation
4) Transfer Function
5) Poles and Zeros
6) Predetermined systems
Signals
7) Help
8) Library
9) Define points
10) Define signals
0) Main menu
Enter your choice (0 . . 10)

5) "System Characteristics" enables the student to analyze the defined systems and to examine their properties, both numerically and graphically:

Numerics
1) Help
2) Matrices
3) Difference Equation
4) Transfer Function
Graphics
5) Help
6) Block Diagram
7) Impulse Response
8) Eigenvalues and Eigenvectors
9) Poles, Zeros and Gain
10) Frequency Response
11) 3-D complex Transfer Function
0) Main menu
Enter your choice (0 . . 11)

6) "Response to an input or initial state" simulates a defined system with an input signal or initial state that can be specified by the student.

IX. Evaluation of the PC Exercises

Since it was the main purpose of the network to improve didactically the exercises for the students, it was important to ask the student for his/her opinion. After each session the student gets the opportunity to fill in a form anonymously; the student can tell whether he/she totally agrees, agrees, has no opinion about, does not agree or totally disagrees with some statements. So far more than 1000 forms are completed. The answers to the most relevant questions are discussed.

The majority of the students (totally) agrees that they understand the theory better after they have done the exercises. On the questions if the programs are easy to use and fast enough, opinions are divided. Here we have a certain duality: the easier the programs are to use, the slower they are and the faster we make the programs, the harder they are to use. So for each session we have tried to make the best compromise between speed and user friendliness. Most students find the presence of an assistant necessary, both for problems concerning the program (or the network) and the subject matter. Depending on the session 60–85% of the students (totally) agrees that a PC session is more interesting than a classic ex cathedra exercise session; 0–5% totally disagrees.

The available software, especially PC-Matlab, is also frequently used by engineering research assistants for research. Furthermore, each year approximately ten students perform the research and write their engineering thesis on the network.

As far as manpower is concerned, most of the design and programming work is performed by recent graduates either full time or half time. It is estimated that the preparation of one session requires about three to six man-months. In total, more than 15 man-years have been spent.

X. Conclusion

The realization of a network of 20 personal computers is a success. It is intensively used by the students for exercise sessions and research; their reactions are very positive. Visitors of the network or participants at conferences [14], [15] are enthusiastic.

Although substantial financial and human efforts already have been made, there is still work to be done: expansion of the hardware (more PC's, memory capacity and printing facilities, possibly some IBM Systems 2), adding new software like a server operating system that is compatible with all the available software (and also operates on a System 2) and programming new exercise sessions. The design and effective operation of the didactical PC network involves many more details than those discussed in the paper. However, the authors are happy to respond to specific questions of the readers and welcome visitors. Indeed, the best way to appreciate the effectiveness is by visiting the PC net and performing the sessions. Moreover, noncommercial software can be mutually exchanged freely.

References

[1] *3Plus Administrator's Guide*. Mountain View, CA: 3Com Corp., 2788-00, rev. 1.1, 1986.
[2] *Micro-Logic II, Digital Design and Simulation Program*. Sunnyvale, CA: Spectrum Software, 1987.
[3] C. Moler, J. Little, and S. Bangert, *PC-Matlab for MS-DOS Personal Computers*. Sherborn: The MathWorks, version 3.13, 1987.
[4] *Gem Desktop*, 3rd ed. Monterey, CA: Digital Research Inc., 1985.
[5] *EtherShare PC Administrator's Guide*. Mountain View, CA: 3Com Corp., version 2.4, Mountain View, 1984.

[6] H. Koçak, *Differential and Difference Equations through Computer Experiments, With Diskettes Containing Phaser: An Animator/Simulator for Dynamical Systems for IBM Personal Computers.* New York: Springer-Verlag, 1986.
[7] B. C. Kuo, *Automatic Control Systems*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[8] F. J. Hill and G. P. Peterson, *Switching Theory and Logical Design, 3rd ed.* New York: Wiley, 1981.
[9] S. Wilson, R. Hale, and C.-D. Hsu, "An inexpensive demonstration of PLL principles," *IEEE Trans. Educ.*, vol. E-23, pp. 200-204, Nov. 1980.
[10] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing.* Boston, MA: Cambridge Univ., 1987.
[11] *Thurlby LA-160 Logic Analyzer.* Huntingdon: Thurlby Electronics, version 2.2, 1986.
[12] K. S. Shanmugan, *Digital and Analog Communication Systems.* New York: Wiley, 1979.
[13] R. A. Gabel and R. A. Roberts, *Signals and Linear Systems.* New York: Wiley, 1973.
[14] D. Reniers *et al.*, "Software for educative purposes in control theory," Benelux Meet. Syst. Contr., Jan. 1987, p. 194.
[15] D. Reniers, J. Vandewalle, and B. De Moor, "A didactical project with a personal computer network for engineering education," Preprints 4th IFAC Sympo. Comput. Aided Design Contr. Syst., CADCE 1988, Beijing, Aug. 25-28, 1988, pp. 372-376.

Joos Vandewalle (S'71-M'79-SM'82) was born in Kortrijk, Belgium, on August 31, 1948. He received the degree of electro-mechanical engineer in electronics in 1971, Doctor in applied sciences in 1976, and Special Doctor in applied sciences in 1984, all from the Katholieke Universiteit, Leuven, Belgium.

He was a Research Associate (1976-1978) and Visiting Assistant Professor (1978-1979) at the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. He was Assistant Professor (1980-1986) and since 1986 he has been a Full Professor (Gewoon Hoogleraar) at the Department of Electrical Engineering, Katholieke Universiteit, Leuven. Since October 1980 he has been a member of the staff of the ESAT Laboratory, K.U. Leuven and since 1985 consultant at IMEC, Leuven. His research interests are mainly in mathematical system theory and its applications in circuit theory, signal processing, cryptography, and neural networks. He has authored or coauthored more than 100 papers in these areas.

Dirk Reniers received the degree in electrical engineering from the Katholieke Universiteit, Leuven, Belgium, in 1986.

He worked at the University of Leuven from 1986 to 1988 where he was responsible for the Personal Computer Network of the ESAT Laboratory. Currently, he is with the Research and Development Department of Volvo Car B.V., Belgium, where he works on electronic control of a continuously variable transmission.

Bart De Moor (S'86-M'88) was born in Halle, Belgium, on July 12, 1960. He received the degree in electrical engineering (control theory) in July 1983 and a doctorate in applied sciences in June 1988, both from the Department of Electrical Engineering, Katholieke Universiteit, Leuven, Belgium.

From September 1988 until September 1989 he was a visiting research associate with the Department of Computer Science (Numerical Analysis Group) and the Department of Electrical Engineering (Information Systems Lab) of Stanford University, Stanford, CA. In October 1989 he became a Research Associate of the Belgian National Fund of Scientific Research at the ESAT Laboratory. Since October 1990 he has been an Assistant Professor at the Katholieke Universiteit, Leuven, Belgium. His research interests include system theory and identification, numerical linear algebra, and complementarity problems with special emphasis towards applications.

Dr. De Moor is a member of the Society for Industrial and Applied Mathematics.