Integrated Learning Platform for Internet-Based Control-Engineering Education

Carlos Andrés Ramos-Paja, *Student Member, IEEE*, José Miguel Ramírez Scarpetta, *Member, IEEE*, and Luis Martínez-Salamero, *Member, IEEE*

***Abstract*—This paper describes an integrated learning platform intended to support Internet-based control-engineering education. Four environments have been developed, namely, remote experimentation, mathematical analysis, dynamic simulation, and self-learning. The learning platform is a virtual support for problem-based education in control engineering. The reported development allows multiple users the access to concepts and both experimental and computational resources through an intuitive and powerful user interface that only requires a Web browser, a low-bandwidth Internet connection, and low user-side technical specifications. The server side is scalable and reproducible since it is based on general public license (GPL) Linux operating system and a Matlab license. The integration of the proposed theoretical, computational, and experimental tools improves previous experiences in Internet-based control education.**

***Index Terms*—Calculation through Internet, concept maps, control-engineering education, dynamic simulation, mathematical analysis, problem-based learning (PBL), remote laboratory.**

I. Introduction

IN RECENT YEARS, the engineering education has experimented multiple changes as a consequence of technological advances in different research fields and new challenges in industry. These changes have required the reorganization of academic curricula and looking for new teaching models for multidisciplinary and problem-oriented applications. This topic has been addressed in different ways. Thus, the experimental-oriented methodology proposed by Ghone *et al.* [1] using multidisciplinary laboratories for mechatronics learning support develops industrial-inspired experimental stations. In a different approach, Munz *et al.* focuses on motivating students through educational games to illustrate the importance and principal concepts of the automatic control engineering [2]. One of the most promising methodology that cover these aspects is the problem-based learning (PBL) [3], whose implementation requires fast and easy access to information, simulation tools, and experimentation platforms. The technological support for this methodology has been addressed in different and separate developments using Web-based platforms for theoretical contents, calculation, simulation, and experimentation.

In online theoretical-content presentation, the common solutions used are the commercially available Blackboard [4] and the open-source platform Moodle [5]. For calculation and simulation, there exist commercial toolboxes that allow online access to traditional platforms like Matlab [6] and Maple [7]. In addition, there are user-developed applications like the online simulation environment reported by Donzellini and Ponta [8] for e-learning of digital design.

Remote experimentation approach has been intensively used in engineering education [9], [10]. This can be observed in the remote laboratory for teleoperation of robotic systems reported by Safaric *et al.* [11] or the multimedia interface to control robot arms presented by Marin *et al.* [12]. These developments allow simulation of algorithms and remote test in real prototypes, interacting through live video and audio with the experiment. In automatic-control-engineering basic learning, Hercog *et al.* developed a DSP-based remote-control laboratory [13], and in the same way, Huba and Simunek presented a PID control teaching environment supported by remote experimentation and Moodle contents [14]. For complex control-engineering learning and industrial application test, Hassan *et al.* reported a powerful environment for remote simulation and experimentation [15].

The available Internet-based control-engineering environments provide information, simulation tools, and experimental platforms, but its design considerations change according to the learning methodology and objectives. In this way, the Internet-based platform proposed by Wu *et al.* [16] provides detailed information to the student about the available experiments and basic control theory. This platform also provides problem simulations to test the control design and access to real experimentation systems, where predefined control structures can be parameterized. This learning platform uses open-source software, which decreases the implementation economic cost. Similarly, Callaghan *et al.* report a control-oriented remote-experimentation environment [17] that also provides theoretical concepts and experiments documentation focused in the learning process. Other developments in this field are based

Manuscript received January 22, 2008; revised January 29, 2009; accepted January 11, 2010. Date of publication February 17, 2010; date of current version September 10, 2010. This work was supported in part by the Grupo de Investigación en Control Industrial (GICI) of the Universidad del Valle, by the Grupo de Investigación en Tecnologías Aplicadas (GITA) of the Universidad Nacional de Colombia—Sede Medellín, by the Automatic Control and Industrial Electronic Group (GAEI) of the Universitat Rovira i Virgili, by the Colombian Departamento Administrativo de Ciencia, Tecnología e Innovación (Colciencias) under the Project 1106-11-17707, and by the Spanish Ministerio de Ciencia e Innovación under the Project TEC2009-13172 and the FPI Scholarship BES-2006-11637.

C. A. Ramos-Paja is with the Facultad de Minas, Universidad Nacional de Colombia—Sede Medellín, Medellin, Colombia (e-mail: caramosp@unal.edu.co).

J. M. Ramírez Scarpetta is with the Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Ciudad Universitaria—Cali, Colombia (e-mail: jomiram@univalle.edu.co).

L. Martínez-Salamero is with the Departamento d'Enginyeria Electrònica, Elèctrica i Automàtica, Escola Tècnica Superior d'Enginyeria, Universitat Rovira i Virgili de Tarragona, 43007 Tarragona, Spain (e-mail: luis.martinez@urv.cat).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2010.2043033

on commercial software. For instance, it has to be pointed out the work developed by Hu *et al.* [18] using Matlab to perform real-time control of the experimental systems and also providing Simulink models of the available prototypes in order to allow the student to perform simulations, therefore requiring a Matlab/Simulink license for that procedure. Another example was found in the remote laboratory for automatic control proposed by Leva and Donida [19], which uses LabVIEW and its Web service to provide a remote-laboratory experience. This commercial-software-based approach is also observed in the remote-control-engineering laboratory reported by Lazar and Carari [20].

Similarly, there exist developments for remote laboratories in other engineering areas like industrial communication networks [21], digital electronic systems [22]–[24], embedded system programming [25], discrete-event systems [26], robotics programming [27]–[30], power electronics [31]–[33], networking engineering [34], and engineering measurement [35].

A common characteristic of these developments is that they do not provide an integrated learning environment, forcing the student to change constantly between platforms and user interfaces, causing, in this way, alterations in the learning process. Moreover, the reported developments do not provide a generic mathematical analysis and open simulation tools, accessible through Internet, to support Web-based control-learning approaches.

In this paper, an integrated learning platform designed to support the PBL methodology in automatic control engineering is presented. The platform provides theoretical-content presentation, generic and open mathematical analysis, and simulation tools, and also gives access to remote experimentation. The main characteristic of this paper is the integration, in a friendly and comprehensible Web-based platform, of the required tools to support the control-engineering education in a nontraditional way for a multidisciplinary field such as automatic control engineering. In addition, the generic mathematical analysis and simulation tools improve the Internet-oriented learning experience.

The rests of this paper are organized as follows. In Section II, an introduction to the problem-based methodology is given and illustrated with some applications in control-engineering education. Next, in Section III, the integrated learning platform and its tools are described. Finally, the conclusions of this paper are given in Section IV.

II. PBL in Automatic-Control-Engineering Education

Automatic control engineering is a multidisciplinary area whose aim is modeling, analysis, and design of automatic regulation systems. The first aspect makes reference to obtain a mathematical or linguistic representation of a system behavior suitable for analysis and simulations. The second one allows establishing relations among variables and elements and their behavior or tendency in time or frequency domains. The last one is related to the definition of rules and algorithms to control the behavior of the variables according to given specifications. The automatic-control-system theory can be used with disciplines like physics, mathematics, mechanics, chemistry, electronics, etc., and even in other fields like sociology, psychology, economy, etc.

In real applications, it must be also taken into account the implementation technology, the compatibility with previous technologies, the cost of the proposed solution, the social benefits of the designed strategy, the execution time of the project, and other aspects like staff management, industrial safety protocols, multidisciplinary teamwork conformation, etc. To summarize, in solving automatic-control-engineering problems, there are many elements that do not exclusively belong to the control-system theory.

New pedagogic models are intended to instruct engineers in the currently required skills to work in multidisciplinary teams, where each person contributes to the group with his/her specialty in the solution of practical problems, which requires high communicative and cooperative aptitudes. In a PBL methodology, several technological and pedagogical tools allow the students to solve real problems in a self-study approach. PBL has been demonstrated to be effective in multidisciplinary areas because it generates the capacity to associate different concepts in a single problem, which makes the PBL an ideal methodology for the learning of automatic control engineering, which is also a horizontal area in engineering curricula [36], [37].

The technological tools that support the educational process have the potential to increase the communicative possibilities to cover bigger quantity of actors and to simulate scenarios where the student are required to use abilities in the solution of real engineering problems. Most of the current technological developments have arisen from other areas of engineering and have been applied to teaching automatic control engineering with good results. In addition, particular developments have been implemented for constrained conditions such as remote laboratories. This makes evident the requirement to develop new technological tools intended to address the specific conditions of PBL methodology in the automatic-control-engineering education.

The main drawback of the currently available automatic control learning environments resides in the low interaction among the different tools, forcing the student to move into different environments, hindering his/her concentration. Additionally, these tools are expensive and require dedicated computational systems. In this paper, we introduce new technological tools to support PBL methodology with low cost and remote access through Internet. Although the developed platform and tools are designed to support automatic control engineering, they are also suitable for general engineering education.

III. Platform Structure

In order to implement a PBL strategy, four basic aspects have been considered: theoretical concepts, mathematical analysis, simulation tools, and experimentation environments. The integrated learning platform considers these four basic aspects and others designed to support the students' performance evaluation.

The theoretical concepts were organized into concept maps [38], [39], which guide the student to learn the desired concepts in a consequent order with the solution of specific

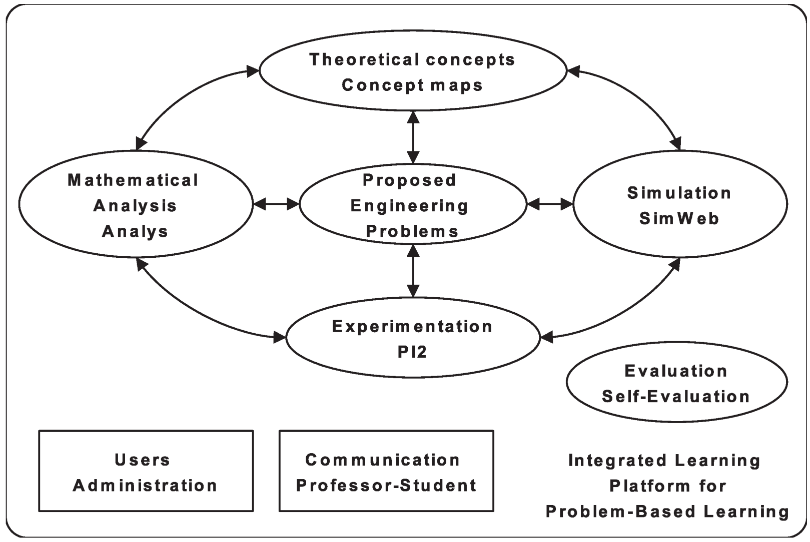

Fig. 1. Integrated learning platform scheme.

designed problems. The user interface of the mathematical analysis and simulation tools were developed using HTML and Javascript languages to allow their execution from the World Wide Web with limited client-side resources. The mathematical calculation and data processing are performed using Matlab (Matlab 7.1 version) through a Transmission Control Protocol (TCP/IP) service. The use of Matlab for mathematical calculation was selected due to its diffusion in the scientific community, calculation precision, and popular toolboxes for different applications. Another interesting option is the open-source Scilab software [40], which allows performing mathematical calculation and provides toolbox for modeling and control applications but requires the development of a communication system.

Fig. 1 shows the integrated learning platform scheme. The learning platform proposes engineering problems to the student in order to stimulate the learning of selected theoretical concepts and the execution of mathematical analysis, simulations, and experimentation. The theoretical-concept management system, evaluation and self-evaluation systems, mathematical analysis and simulation tools, experimentation environment, and user-administration systems are described in the next subsections.

A. Theoretical-Concept Management and Concept Maps

Following the PBL methodology, the theoretical concepts were organized according to its use in the solution of engineering problems and practical projects. The pedagogic tool selected for the theoretical concepts organization was *concept maps*, which allow the relational organization of information in a simple, intuitive, and comprehensible structure, where each concept has a previous-knowledge requirement in order to understand the information presented. In that way, the concept maps guide the student through the minimum required information to understand a particular concept in order to use it in the solution of practical problems. An example of concept mapping is given in Fig. 2, where a learning process for continuous dynamic-system modeling is proposed. The different concepts selected to obtain a mathematical model of a multivariable linear system in transfer-function or state-space representations can be observed. The concept maps are presented to the student using interactive Web graphics that allow an intuitive and fast access to the concepts through Internet. Like the concept map shown in Fig. 2, they allow the student to access any concept by pressing on it. These concepts are theoretically explained, illustrated with interactive animations, and evaluated.

The platform infrastructure is flexible, and it can be used in different approaches as the one shown in the flowchart of Fig. 3. First, an engineering problem is proposed, and to solve it, the student must review theoretical concepts and use the mathematical, simulation, and experimentation tools. Some of the proposed engineering problems are tank-pressure control, water-temperature control, tank liquid-level control, and the simultaneous control of water level and temperature in a multivariable process.

The concepts are explained using hypertext and Macromedia Flash animations [41], and the students are allowed to browse different topics using links among them. In addition, the concepts are complemented with suggestion to simulation or experimentation and with self-evaluation tests. Fig. 4(a) shows the concept tool Web site map, where the login system page that gives access to the platform main menu can be observed and from which it is possible to start any tool (concept management, mathematical analysis and simulation, and experimentation). The concept tool gives access to interactive concept maps and proposed engineering problems, which give a learning objective to the student. The user interface of the concept tool, shown in Fig. 4(b), provides a contextual menu that gives access *in situ* to the mathematical analysis and simulation tools. Additionally, from this menu, it is possible to review the proposed engineering problems and interactive concept maps, which permits accessing any desired concept explanation. The information is exposed to the student in a work zone. Moreover, a map route and a recommended map-navigation path are given.

The mathematical analyses are performed by using a specific control-oriented *Analys* component developed to make transparent to the student the Matlab commands, but it is also

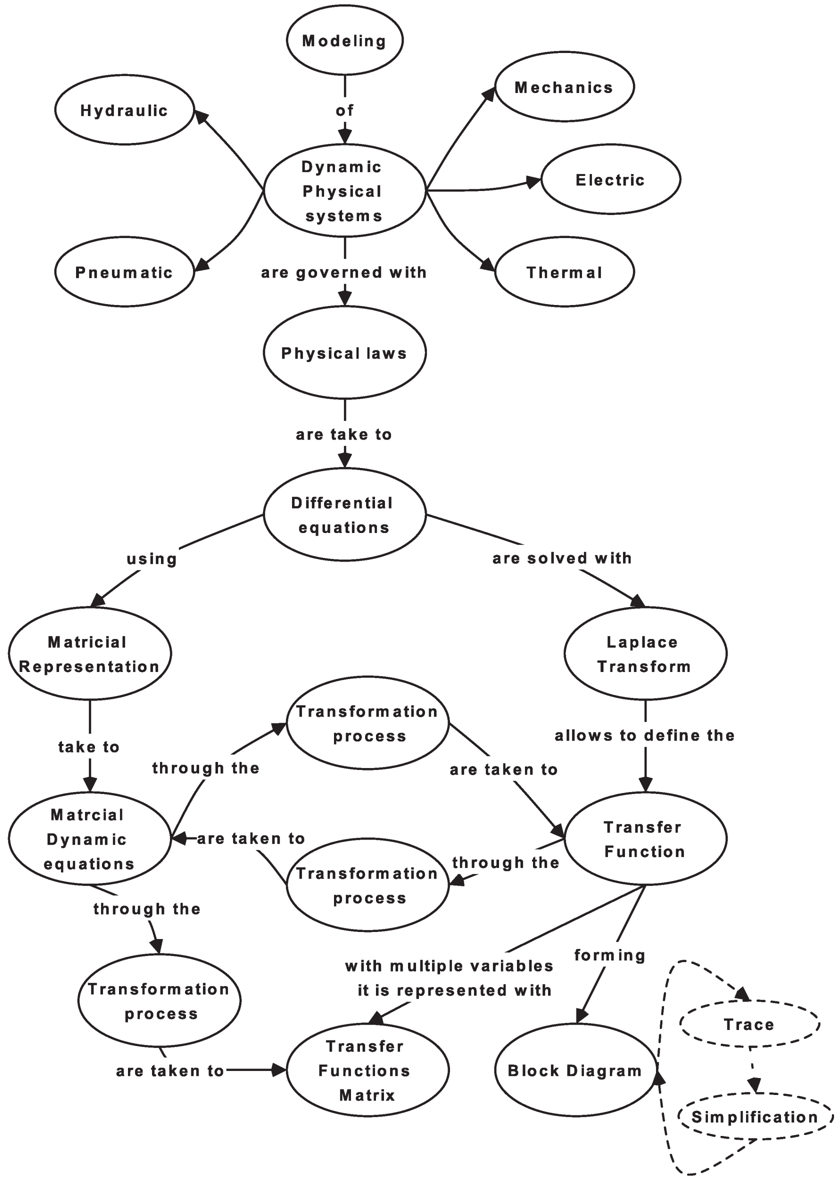

Fig. 2. Concept map for modeling of continuous dynamic systems.

available as a command-based application that gives access to Matlab commands, allowing the performance of complex and user-designed analyses. The simulation task is performed by using a generic Web-based dynamic simulator *SimWeb*, which allows the in-platform design of the block diagram intended to be simulated. Finally, the experimentation task is performed by using the Web interface *PI2*, which allows execution, inspection, and collection of data from predefined experiments.

The relations, connections, and evaluations are stored in a PostgreSQL [42] database, and they are also managed by a system implemented using the Apache Web server, and the hypertext preprocessor (PHP) active page service [43]. The operating system used in the server is the Community ENTerprise Operating System Linux due to its excellent performance and GPL.

The structure of the concept maps and evaluation-management and user-administration systems are shown in Fig. 5. The users access the learning platform using conventional Web browsers through the Apache Web server. Next, the active pages developed using the PHP language use the data stored in the PostgreSQL database in order to administrate the users' access, display the concept information and connections following the defined concept maps, present the self-evaluation tests, and perform the evaluation of the student responses. In addition, depending on the student qualification obtained in the tests, the system suggests the review of previous concepts or the access to the next one.

In the integrated learning platform, the instructors can use the default problems and tests available or can define specific ones for different groups of students. In the same way, the tests can be timed or not, thus allowing evaluation of the student skills including his/her response time. The learning platform stores the student responses to the test questions and allows some basic but powerful analyses: performance of specific student in a concept or set of concepts, performance of student groups in a concept or in a particular question, etc. Additionally, the system allows downloading the student responses and qualifications in order to perform a more sophisticated statistical analysis with specialized software.

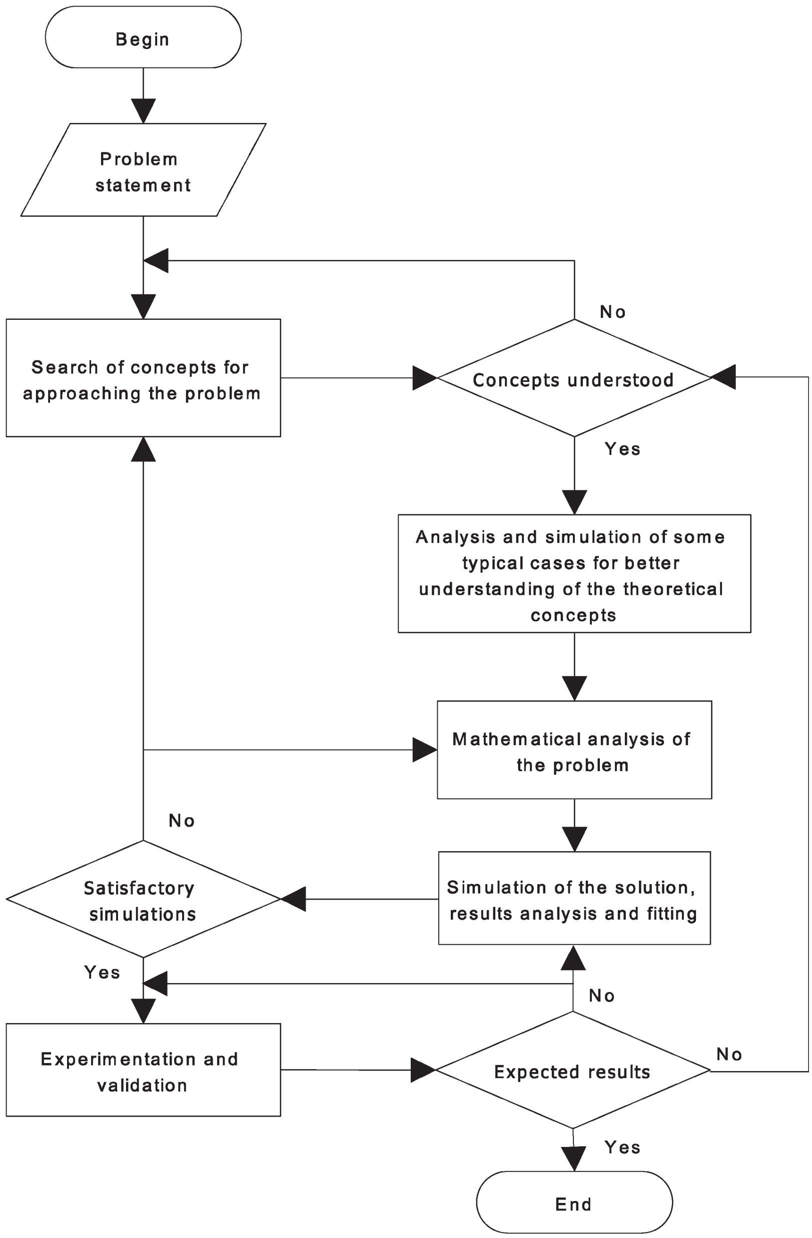

Fig. 3. Interactive-learning flowchart.

The user-management system has three main objectives: the first one is to control the access to the learning platform; due to the limited nature of the resources, some access priorities must be taken into account. The second objective is to store the students' performance information to be analyzed. The final task is to schedule the access to the remote experimentation platform to avoid physical conflicts.

To summarize, the integrated learning platform allows the implementation of concept maps in order to provide the students an interactive learning experience and gives the possibility to track the students' performance and identify collective difficulties.

B. Mathematical Analysis and Simulation Web Tools

In order to provide in-platform and remote mathematical analysis and simulation, two Web-based tools have been developed. The mathematical analysis tool *Analys* and the simulation tool *SimWeb* have been designed to be accessible through Internet using a low-bandwidth connection. The structure of the mathematical calculation Web tools is shown in Fig. 6.

The Web tools were developed using HTML, Javascript, PHP and Matlab languages. The user interfaces were developed using HTML and Javascript languages, they being based on forms, tables, buttons, and graphics. The PHP language was used to develop the interface between the process and calculation engines and the Web-based user interfaces. The data-processing algorithms for the analyses and simulations were developed using Matlab language (m-functions) and its ToolBox functions.

To execute a user analysis or simulation request, the user interface codifies the information, which is transmitted to the process engine and through a TCP/IP service to the Matlab engine using the PHP active pages. The process engine uses the

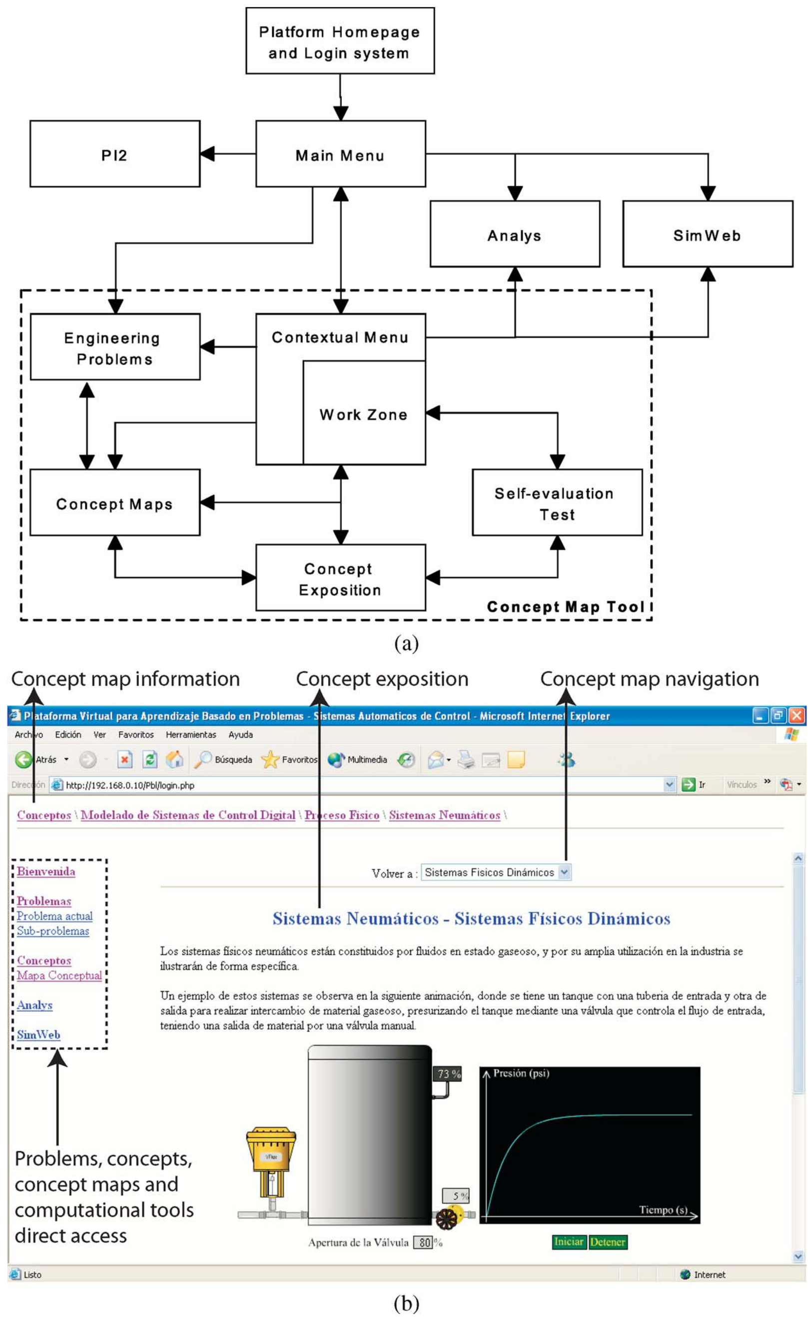

Fig. 4. Concept map tool. (a) Web site map. (b) User interface.

Matlab engine to perform the desired calculations and stores the results in tables, graphics, and data files in the server. Later, the process engine and the TCP/IP service call the PHP active pages to return the results to the user, displaying the requested data calculations, graphics generated, and links to the data files.

The TCP/IP service that allows the communication between the Web user interfaces and the Matlab engine has been developed by means of the Matlab Java Runtime Class, which permits starting a new Matlab process, sends commands to this process, and receives results back [44]. Using the Matlab Java interface, a Java server application residing in the server side, which process the command request, has been developed. This Java server interacts with the PHP active pages in order to receive the request from the user interface and also to send the results, which are text or graphics-based information. This communication system and the Java-based Matlab interaction strategy allow supporting multiple users by means of a single Matlab license. This is also allowed by the possibility of starting simultaneous Matlab instances (and process) in the selected Linux operating system, each one of them with its own workspace, which is closed when the requested process has been performed.

The main characteristics of the Web-based mathematical analysis and simulation engine are as follows.

1) The mathematical models are coded in HTML forms, and the results are given in Web pages and JPEG graphics in order to generate a low-bandwidth consumption for information transmission.

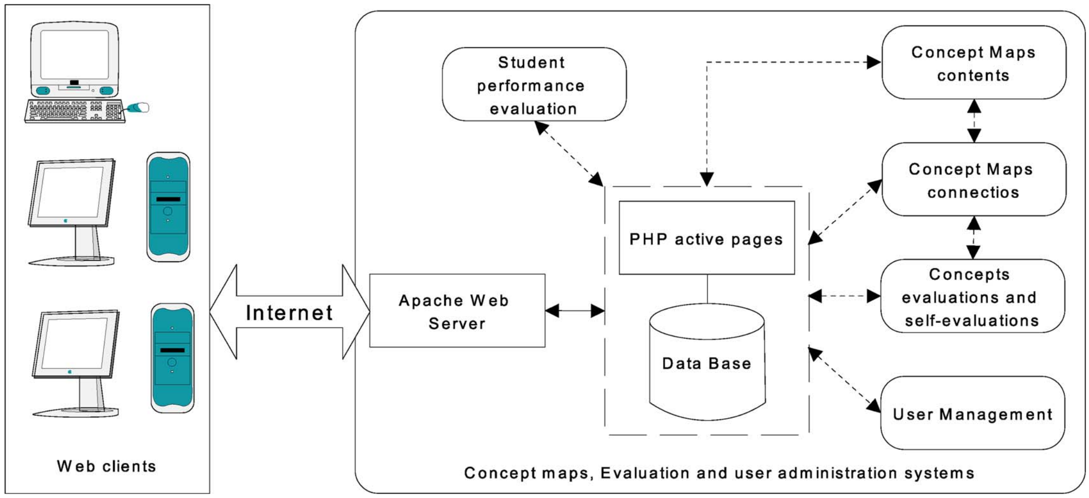

Fig. 5. Concept maps and evaluation-management and user-administration systems.

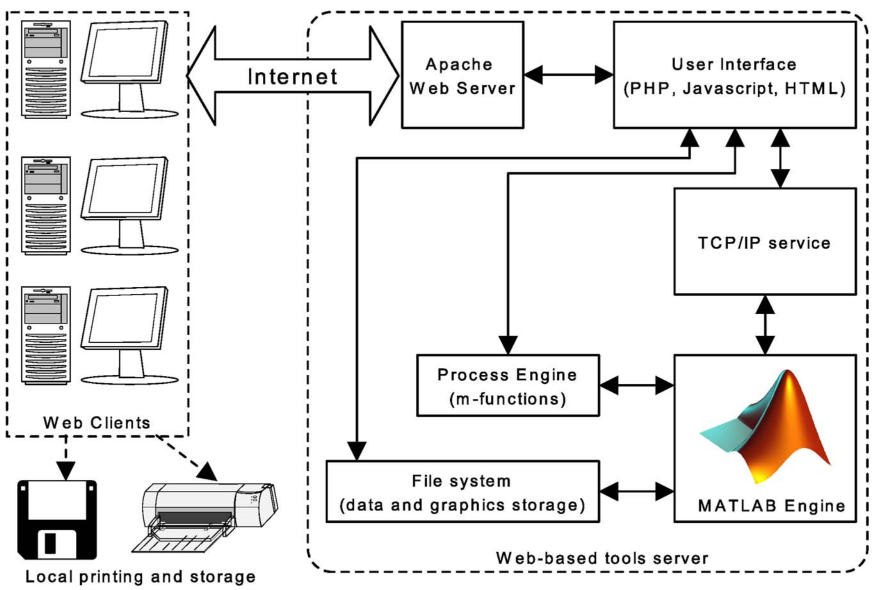

Fig. 6. Web tools structure—*Analys* and *SimWeb*.

2) The updating of the tools is carried out directly in the server, and therefore, the new version is available immediately for all the users of the system.
3) The number of simultaneous users of *Analys* and *SimWeb* is determined by the server performance and not by the number of Matlab licenses available, which implies an important economical reduction.
4) The engine is easily scalable because it is possible to use several Matlab and additional Apache and PHP servers.
5) The analysis and simulation results are also given in downloadable graphic and data files for local printing and storage in order to perform further analysis.

Finally, this Web-based mathematical analysis and simulation engine allows an easy and low-cost access to important tools in automatic-control-engineering education. The tools can be used from any geographical location at any time, requiring only a Web browser available in any client operating system.

1) Mathematical Analysis Tool—Analys: Analys is a mathematical analysis tool for multivariable linear systems designed to support the automatic-control-engineering learning. In *Analys*, the student defines a mathematical model in state-space, transfer-function, or pole–zero representation. The linear system can be continuous or discrete, and it is possible to carry out the different operations and analyses listed as follows:

1) time analysis: step, impulse, user-defined signal, and initial-state response;
2) frequency analysis: Bode diagram, Nyquist diagram, Nichols chart, and stability margins;
3) state-space analysis: transformation into canonical forms, similarity transformations, observability, and controllability;
4) root-locus analysis;
5) model conversion: continuous to discrete, discrete to continuous, and resampling of a discrete model;
6) dynamics of the system: bandwidth, dc gain, and poles and zeros.

The user interface of *Analys,* shown in Fig. 7, presents the linear model defined by the user in state space and by means

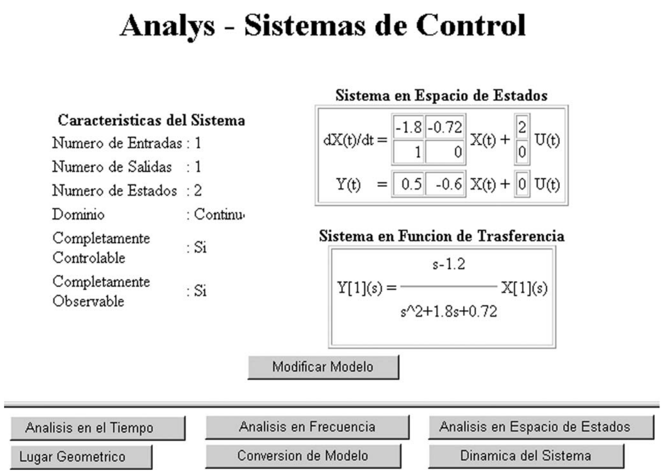

Fig. 7. *Analys* user interface.

of a transfer-function representation and also gives information about the model: number of states, number of inputs and outputs, time domain, absolute controllability, and observability. In the example shown in Fig. 7, the single-input–single-output (SISO) model described by (1) has been loaded into *Analys*. The buttons to perform the analyses and to modify the model can be also seen, i.e.,

$$\dot{x}(t) = \begin{bmatrix} -1.8 & -0.72 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u(t)$$
$$y(t) = [0.5 \quad -0.6]x(t) + [0]u(t). \tag{1}$$

The graphic results to step response and Nyquist diagram analyses performed with *Analys* are shown in Fig. 8(a) and (b), respectively. These results are presented to the user as JPEG graphics in a results page, but another information like vectors or matrices are presented in text-based result pages. In addition, it is possible to download data files of the graphics results for additional analysis as shown in Fig. 8(a), where a link to the data file is provided on the bottom of the result page.

Analys has an additional general-purpose command-line interface that allows execution of Matlab commands, including graphic-manipulation tools and Toolbox functions. This interface is useful for any mathematical analysis, including automatic control systems, where it is possible to use the Toolbox functions installed in the server, this with the exception of commands that have its own graphic user interfaces, i.e., `sisotool()` command. An example of this general-purpose command-line interface is shown in Fig. 8(c), where the following command list is executed:

```
s = tf('s');
G = (s + 1)/(s^2 + 3 * s + 5)
subplot(2, 2, 1)
step(G)
subplot(2, 2, 2)
impulse(G)
subplot(2, 2, 3)
rlocus(G)
subplot(2, 2, 4)
bode(G).
```

In Fig. 8(c), the text-based responses of the Matlab engine can be seen, corresponding in this example to the echo of the transfer function G(s) definition. In addition, the `subplot()` and the Matlab *control system Toolbox* commands request the plotting of the step and impulse responses of G(s) and also its root locus and Bode diagrams, which are depicted in the graphic result page shown in Fig. 8(d).

Analys allows the student to perform iterative procedures using the Matlab programming language by using the command-line interface. *Analys* command-based and client-server structures make possible the design of new user interfaces to support additional control-engineering tasks, i.e., interactive parameter tuning by means of Java or Macromedia Flash interactive capabilities.

To summarize, *Analys* is a Web-based mathematical analysis tool designed to support automatic control-engineering learning but has many features that make it useful in other fields. Moreover, *Analys* has a general-purpose command-line interface that allows the execution of user-designed algorithms and analyses, and it presents text-based, graphic, and data results. In addition, *Analys* technological characteristics allow the design of additional interactive user interfaces to support iterative, tuning, or *in situ* design process.

2) Simulation Tool—SimWeb: SimWeb is a Web-based Simulink-like environment that allows a fast and easy design of block diagrams and its simulation through low-bandwidth Internet connections. *SimWeb* supports the technology *drag-and-drop* standard for block-based simulation applications, such as Simulink from Matlab [6] and Scicos from Scilab [40]. *SimWeb* allows performing simulations of linear, nonlinear, continuous, and discrete systems or combinations of them.

The user interface of *SimWeb* is divided into three zones as shown in Fig. 9(a): block zone, work zone, and configuration zone. In the block zone, the function blocks available can be found. The work zone is the workspace where the block diagram is designed, and the blocks can be dragged from the block zone and dropped in this area. In addition, it is possible to move the blocks placed in this zone using the same *drag-and-drop* functionality. In the configuration zone, the parameters of a selected block are displayed, which can be modified and saved by pressing the *save button*. Other buttons of the configuration zone are the *information button* that open a pop-up window with data about the active block, the *erase button* that delete the active block, the *user-manual button*, the *connection button*, and the *simulation button*.

The connection button calls the *connection utility* that allows administration of the links between blocks. This tool permits defining the blocks that will be connected to the active block inputs. In Fig. 9(b), the connections of the scope block of the example in Fig. 9(a) are shown, where two inputs are defined and the identifier given by the user to the block is specified in red text. In the connection description of Fig. 9(b), the connected input and the identifier of the block attached to that specific input are given, and a disconnection button is also available. In the example, the block identified as *Proceso* is connected to input 1 of the active block *Variables*, and the block named *Muestreador* is connected to the input 2 of the same active block. Additionally, a *new connection section* is observed

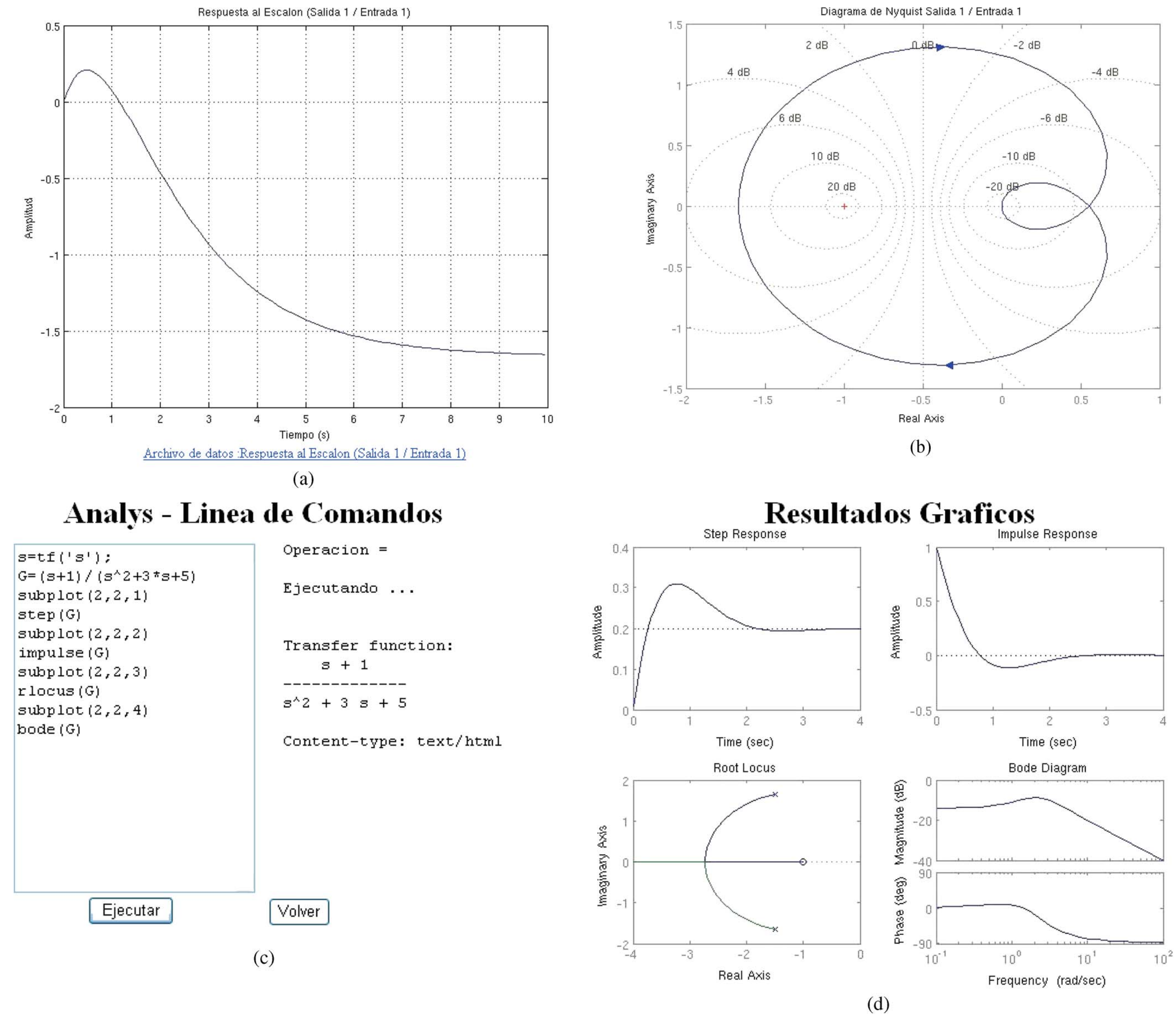

Fig. 8. *Analys* output analysis and command line. (a) Step response. (b) Nyquist diagram. (c) Command line. (d) Graphics results.

in the bottom of the tool, where it is possible to define a new link to one free input from the output-enable block. Finally, when a connection is defined, the *SimWeb* graphical engine traces the visual connection between the blocks in the work zone. Similarly, if a block is moved, the graphical connections will be replotted, and also if a block is erased, the respective connections will be deleted.

The simulation button calls the *coding and sending* algorithm, which compresses and transfers the block-diagram information to the processing and Matlab engines. In this process, the information stored in the connection and parameter tables in the Javascript-based user interface (client side), which defines the block diagram, are transmitted to the PHP active-page engine which uses the Java server to interact with the Matlab engine. The processing engine, developed using Matlab m-functions, decodes the block-diagram information and executes the block code taking into account the user parameter definitions. The block code is also implemented using m-functions, and therefore, the Simulink software (and license) is not required to process the *SimWeb* block diagrams.

The results of the simulations are presented to the user in result pages with the graphics requested in the scope blocks. In the example of Fig. 9(a), a closed-loop transfer function with both time- and frequency-domain scopes is defined. The time-domain scope plots the output of a linear system $\mathtt{G}(\mathtt{s})$ and also a sampled signal of the same output. The frequency-based scope calculates a fast Fourier transformation of the same $\mathtt{G}(\mathtt{s})$ output. The graphic results of the scopes are shown in Fig. 9(c) from left to right, respectively.

In conclusion, *SimWeb* is a Simulink-like simulation environment designed to be used through Internet, requiring only a Web browser and a low-bandwidth connection. *SimWeb* has an intuitive *drag-and-drop* interface with a powerful Matlab engine calculation backend without requiring a Simulink license. Like *Analys*, *SimWeb* provides downloadable graphic and data files of the performed simulations.

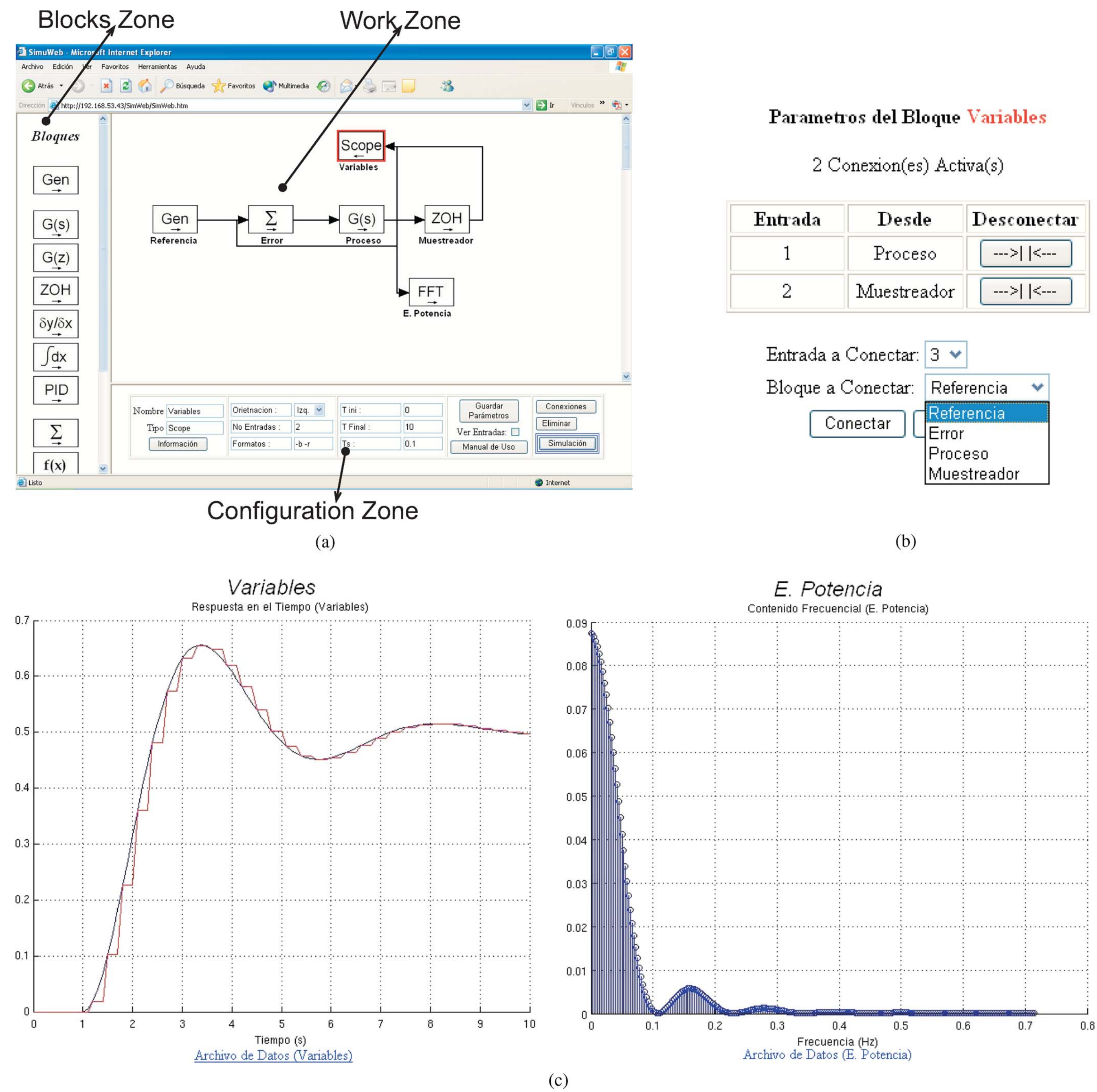

Fig. 9. *SimWeb* user interface and result page. (a) *SimWeb* user interface. (b) Connection utility. (c) Simulation scope output.

C. Remote-Experimentation Environment PI2

Real laboratory prototypes are accessible by the students through Internet in order to provide a complete experimentation environment. The Industrial Informatics Experimentation Platform *PI2* has multiple sensors and actuators that allow configuring different control experiments. *PI2* has a heat exchanger that permits the realization of multiple experiments of heat transfer and fluid dynamics. The platform is complemented with tanks, pumps, and valves that allow the configuration of different types of experiments like liquid-level control, liquid-flow control, gas-pressure control, temperature control, heat-transfer regulation, and any combination of them to obtain multivariable process like cascade-tank-level control. In addition, the actuators allow performing disturbances in the process and fast reconfiguration of the platform for a new experiment. The time constants of the basic experiments, like the flow regulation or fluid-level control, are in seconds, while the thermal experiments (temperature control and heat-transfer regulation) are in decades of minutes. The *PI2* multivariable-experiment time constant depends on the basic experiments involved. The physical structure of *PI2* can be seen in Fig. 10.

PI2 is controlled by a programmable logic controller (PLC) that interacts with the sensors and actuators. This PLC is managed by a *control server* that stores the experimental data

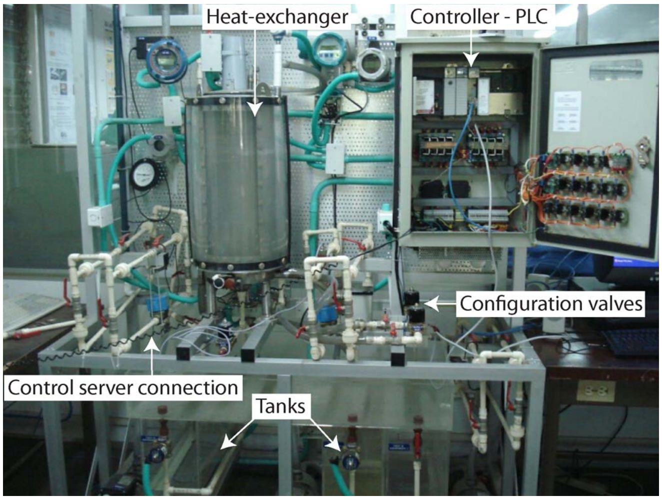

Fig. 10. PI2—Experimentation Platform.

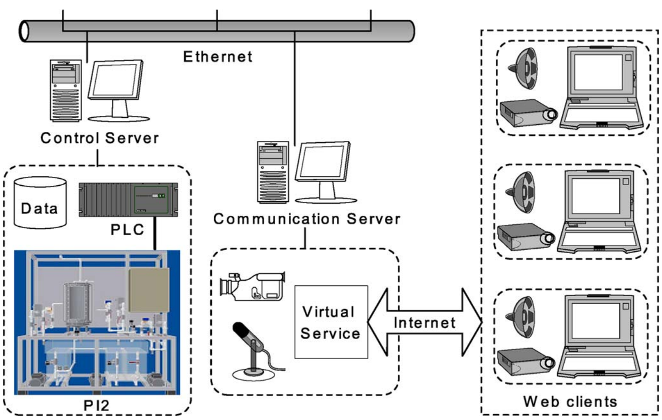

Fig. 11. Remote-experimentation-platform scheme.

in a database and exchange information with a remote user through a *communication server*. This server also provides a multimedia service that allows acquiring video and audio from the experimental platform in real time and presents it to the user. This experimental experience is accessible through Internet due to a virtual service, which allows defining the experimental configuration and control parameters and observing the process signals, responses, and physical platform in both live video and audio. Moreover, the PLC programming implements protection routines to avoid hardware damages caused by erroneous control designs by shutting down the experiment when a dangerous situation is detected. A scheme diagram of the remote-experimentation platform is shown in Fig. 11.

PI2 has an open structure where additional control and communication servers can be included to increase the number of industrial processes and services in order to enhance the experimental environment and allow the simultaneous access of multiple users. The remote access to *PI2* provides a Web-based experiment-execution environment and a multimedia interaction through Internet. Fig. 12 shows two tabs of the *remote user interface* of *PI2*, where real-time audio and video, interactive animations of the process, and real-time graphics of the variables are shown, and also where control parameters can be modified by the user.

When the user selects a desired experiment in the remote interface, the *control server* sets the appropriate valve and pump configurations and load the selected control strategy in the PLC. Later, through the *communication server*, the multimedia and control data is accessed and modified by using a Java-enabled Web browser. In the same way with *Analys* and *SimWeb*, this experimentation tool provides downloadable data files of the performed experiments. The *PI2* platform has an additional system to administrate the users' access to the physical resources. This is a database-management system that gives time access to the users in order to perform a defined sequence of experiments. Users, their schedule, and the available experiments for each of them are stored by the instructor using a Web-based administrative tool. This system is indispensable because the physical resources cannot be shared in the execution of experiments, and therefore, two users cannot perform experiments simultaneously. The user access can be scheduled from Monday to Saturday at any time of the day, excepting between 8:00 AM and 9:00 AM, when the laboratory operator verifies the experimental-platform integrity and repair it when necessary. In addition, the *PI2* scheduled system leaves 5 min interval between experiments in order to avoid hardware stress.

To summarize, *PI2* is a powerful and versatile remote-process-control laboratory, which allows performing multiple SISO and multiple-input–multiple-output (MIMO) control experiments through Internet. This makes *PI2* an important tool in the automatic-control-system learning and one of the main axes of the proposed integrated learning platform. On the other hand, in conjunction with *Analys* and *SimWeb*, it provides a complete analysis, simulation, and experimentation environment to support remote and PBL of automatic control engineering.

IV. Conclusion and Further Work

The presented integrated learning platform offers multiple benefits to guided and autonomous learning process. The students have a wide range of methodological, mathematical analysis, and simulation tools, and a complete experimental experience, which allow approaching different themes from a remote workstation through a low-bandwidth Internet connection and a Web browser. These features give access to technological tools and theoretical concepts, and together with the instructor remote communication, allow the student to approach complex problems and to acquire the desired skills in the solution of practical engineering problems.

At the other end, instructors and tutors have tools to track the collective and individual performance of the students in the addressed areas. These advantages allow the tutors to add or modify contents and tests in order to support the student's gradual learning. Similarly, the platform allows defining innovative work methodologies and group projects.

This platform has been used in the automatic control engineering (710151M) and industrial process control (710099M) courses in the Universidad del Valle (Cali, Colombia), since 2007, with an average number of 43 students per year. The access to the platform has intensified the use of the analytical and simulation tools by the students and has also enlarged the autonomous experimental verification of the proposed

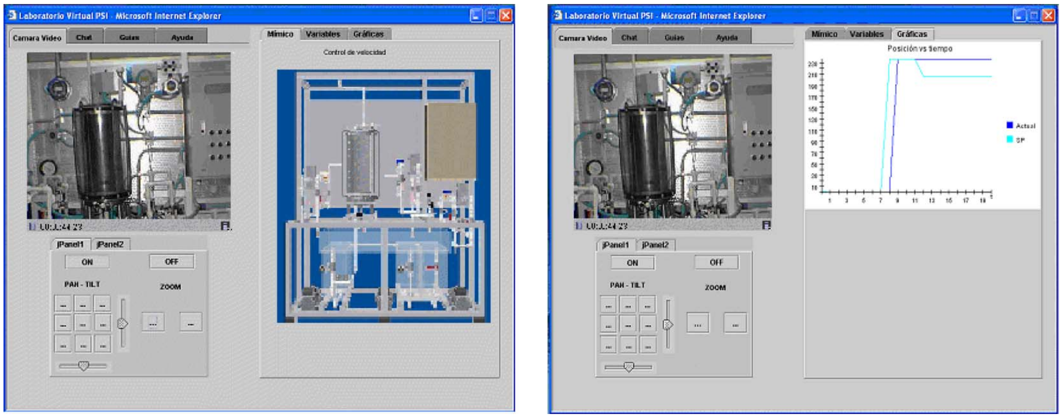

Fig. 12. Remote user interface for PI2.

simulation tasks, this being mainly been caused by the fast and easy access to the resources. In addition, since the students have access to the platform, the amount of questions during the lectures has been significantly increased, and the students' interests for experimental performances has been vigorously enhanced.

This integrated learning platform is intended to support an Internet and *PBL* methodology in the automatic-control-engineering education. This development allows multiple users to access the Matlab resources through an intuitive and powerful user interface. Additionally, it is possible to use the integrated learning system in different forms, obtaining an applicable and powerful tool for different educational and learning methodologies in engineering and related areas.

This PBL platform also guides the student through appropriate concepts, procedures, and experiments to get the desired knowledge in either an autonomous process, a presential form, or through low-cost Internet connection. Thus, the university education coverage is enlarged in both geographical and social aspects.

Finally, it is expected to expand the proposed Internet-based platform to other areas, like digital signal processing. Increasing the mathematical analysis and simulation tools and developing new remote experimental environments to support these new areas is in progress.

REFERENCES

[1] M. Ghone, M. Schubert, and J. Wagner, "Development of a mechatronics laboratory—Eliminating barriers to manufacturing instrumentation and control," *IEEE Trans. Ind. Electron.*, vol. 50, no. 2, pp. 394–397, Apr. 2003.

[2] U. Munz, P. Schumm, A. Wiesebrock, and F. Allgower, "Motivation and learning progress through educational games," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3141–3144, Dec. 2007.

[3] F. Dochy, M. Segers, P. Van den Bossche, and D. Gijbels, "Effects of problem-based learning: A meta-analysis," *Learn. Instruction*, vol. 13, no. 5, pp. 533–568, Oct. 2003.

[4] R. Morales-Menendez, I. Chavez, M. Cadena, and L. Garza, "Control engineering education at Monterrey Tech," in *Proc. Amer. Control Conf.*, 2006, pp. 286–291.

[5] M. Suchanska and J. Keczkowska, "Some aspects of employing the Moodle platform as a tool for enhancing the teaching and learning process," in *Proc. Int. Conf. 'Computer as a Tool'—EUROCON*, 2007, pp. 2465–2467.

[6] MathWorks, Mathworks Home Page2007. [Online]. Available: http://www.mathworks.com

[7] MapleSoft, Maplesoft Home Page2007. [Online]. Available: http://www.maplesoft.com

[8] G. Donzellini and D. Ponta, "A simulation environment for e-learning in digital design," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3078–3085, Dec. 2007.

[9] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.

[10] R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido, "Development of a web-based control laboratory for automation technicians: The three tank system," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 35–44, Feb. 2008.

[11] R. Safaric, M. Debevc, R. Parkin, and S. Uran, "Telerobotics experiments via Internet," *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 424–431, Apr. 2001.

[12] R. Marin, P. Sanz, P. Nebot, and R. Wirz, "A multimodal interface to control a robot arm via the web: A case study on remote programming," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1520, Dec. 2005.

[13] D. Hercog, B. Gergic, S. Uran, and K. Jezernik, "A DSP-based remote control laboratory," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3057–3068, Dec. 2007.

[14] M. Huba and M. Simunek, "Modular approach to teaching PID control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3112–3121, Dec. 2007.

[15] H. Hassan, C. Dominguez, J.-M. Martinez, A. Perles, and J. Albaladejo, "Remote laboratory architecture for the validation of industrial control applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3094–3102, Dec. 2007.

[16] M. Wu, J.-H. She, G.-X. Zeng, and Y. Ohyama, "Internet-based teaching and experiment system for control engineering course," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2386–2396, Jun. 2008.

[17] M. Callaghan, J. Harkin, T. McGinnity, and L. P. Maguire, "Intelligent user support in autonomous remote experimentation environments," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2355–2367, Jun. 2008.

[18] W. Hu, G.-P. Liu, D. Rees, and Y. Qiao, "Design and implementation of web-based control laboratory for test rigs in geographically diverse locations," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2343–2354, Jun. 2008.

[19] A. Leva and F. Donida, "Multifunctional remote laboratory for education in automatic control: The CrAutoLab experience," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2376–2385, Jun. 2008.

[20] C. Lazar and S. Carari, "A remote-control engineering laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.

[21] L. Lo Bello, O. Mirabella, and A. Raucea, "Design and implementation of an educational testbed for experiencing with industrial communication networks," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3122–3133, Dec. 2007.

[22] L. S. Indrusiak, M. Glesner, and R. Reis, "On the evolution of remote laboratories for prototyping digital electronic systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3069–3077, Dec. 2007.

[23] W. Pleskacz, V. Stopjakova, T. Borejko, A. Jutman, and A. Wakanis, "DefSim: A remote laboratory for studying physical defects in CMOS digital circuits," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2405–2415, Jun. 2008.

[24] A. Mittal, C. Gupta, and A. Gupta, "Addressing the bandwidth efficiency, control, and evaluation issues in software remote laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2326–2333, Jun. 2008.

[25] D. Lopez, R. Cedazo, F. Sanchez, and J.-M. Sebastian, "Ciclope robot: Web-based system to remote program an embedded real-time system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4791–4797, Dec. 2009.

[26] P. Marange, F. Gellot, and B. Riera, "Remote control of automation systems for DES courses," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3103–3111, Dec. 2007.

[27] J. Fernandez, R. Marin, and R. Wirz, "Online competitions: An open space to improve the learning process," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3086–3093, Dec. 2007.

[28] F. Zeiger, M. Schmidt, and K. Schilling, "Remote experiments with mobile-robot hardware via Internet at limited link capacity," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4798–4805, Dec. 2009.

[29] R. Marin, G. Leon, R. Wirz, J. Sales, J. Claver, P. Sanz, and J. Fernandez, "Remote programming of network robots within the UJI industrial robotics telelaboratory: FPGA vision and SNRP network protocol," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4806–4816, Dec. 2009.

[30] A. Balestrino, A. Caiti, and E. Crisostomi, "From remote experiments to web-based learning objects: An advanced telelaboratory for robotics and control systems," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4817–4825, Dec. 2009.

[31] V. Pires, L. S. Martins, T. Amaral, R. Marcal, R. Rodrigues, and M. Crissstomo, "Distance-learning power-system protection based on testing protective relays," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2433–2438, Jun. 2008.

[32] S.-C. Wang and Y.-H. Liu, "Software-reconfigurable e-learning platform for power electronics courses," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2416–2424, Jun. 2008.

[33] R. Marques, J. Rocha, S. Rafael, and J. Martins, "Design and implementation of a reconfigurable remote laboratory, using oscilloscope/PLC network for WWW access," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2425–2432, Jun. 2008.

[34] J. Prieto-Blazquez, J. Arnedo-Moreno, and J. Herrera-Joancomarti, "An integrated structure for a virtual networking laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2334–2342, Jun. 2008.

[35] M. Restivo, J. Mendes, A. Lopes, C. Silva, and F. Chouzal, "A remote laboratory in engineering measurement," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4836–4843, Dec. 2009.

[36] A. Massey, V. Ramesh, and V. Khatri, "Design, development, and assessment of mobile applications: The case for problem-based learning," *IEEE Trans. Educ.*, vol. 49, no. 2, pp. 183–192, May 2006.

[37] S. Hussmann and D. Jensen, "Crazy car race contest: Multicourse design curricula in embedded system design," *IEEE Trans. Educ.*, vol. 50, no. 1, pp. 61–67, Feb. 2007.

[38] J. Turns, C. J. Atman, and R. Adams, "Concept maps for engineering education: A cognitively motivated tool supporting varied assessment functions," *IEEE Trans. Educ.*, vol. 43, no. 2, pp. 164–173, May 2000.

[39] S.-S. Tseng, P.-C. Sue, J.-M. Su, J.-F. Weng, and W.-N. Tsai, "A new approach for constructing the concept map," *Comput. Educ.*, vol. 49, no. 3, pp. 691–707, Nov. 2007.

[40] Scilab, Scilab Home Page. [Online]. Available: http://www.scilab.org

[41] R. Rubio Garcia, J. Suarez Quiros, R. Gallego Santos, S. M. González, and S. Moran Fernanz, "Interactive multimedia animation with Macromedia Flash in descriptive geometry teaching," *Comput. Educ.*, vol. 49, no. 3, pp. 615–639, Nov. 2007.

[42] D. Egan, P. Zikopoulos, and C. Rogers, "PostgreSQL on Linux," in *DBAs Guide to Databases Under Linux*. Burlington, MA: Syngress, 2000, pp. 359–418.

[43] R. Stone, "Validating scripted web-pages," *Electron. Notes Theor. Comput. Sci.*, vol. 157, no. 2, pp. 193–205, May 2006.

[44] A. Klimke, How to Access Matlab From Java2003. [Online]. Available: http://preprints.ians.uni-stuttgart.de/downloads/2003/2003-005.pdf

Carlos Andrés Ramos-Paja (S'08) received the B.S. degree (with honors) in electronics engineering and the M.Sc. degree in automatic control from the Universidad del Valle, Cali, Colombia, in 2002 and 2005, respectively, and the Ph.D. degree (with honors) from the Universitat Rovira i Virgili de Tarragona, Tarragona, Spain, in 2009.

He is currently an Assistant Professor with the Universidad Nacional de Colombia—Sede Medellín, Medellin, Colombia. His main research interests include design and control of power systems for renewable energy sources.

José Miguel Ramírez Scarpetta (M'98) received the B.S. degree from Universidad del Valle, Cali, Colombia, in 1986, the M.Sc. degree in electrical engineering in 1988, and the Ph.D. degree from the Institut National Polytechnique de Grenoble, Grenoble, France.

He has been a Professor with the School of Electrical and Electronic Engineering, Universidad del Valle, since 1988. He is the Director of Industrial Control Research Group. He is the holder of an international patent on minimum energy control for induction motors. His main research fields are optimal and nonlinear control of electromechanical systems and engineering education.

Luis Martínez-Salamero (S'77–M'85) received the Ingeniero de Telecomunicación and the Ph.D. degrees from the Universidad Politécnica de Catalunya, Barcelona, Spain, in 1978 and 1984, respectively.

From 1978 to 1992, he taught circuit theory, analog electronics and power processing with Escuela Técnica Superior de Ingenieros de Telecomunicación de Barcelona. During the academic year 1992–1993, he was a Visiting Professor at the Center for Solid State Power Conditioning and Control, Deparment of Electrical Engineering, Duke University, Durham, NC. During the academic year 2003–2004, he was a Visiting Scholar at the Laboratoire d'Architecture et d'Analyse des Systémes (LAAS), Research National Center (CNRS), Toulouse, France. He is currently a Full Professor with the Departamento de Ingeniería Electrónica, Eléctrica y Automática, Escuela Técnica Superior de Ingeniería, Universidad Rovira i Virgili, Tarragona, Spain. His research interest are in the field of structure and control of power conditioning systems for autonomous systems. He has published a great number of papers in scientific journals and conference proceedings and is a holder of a U.S. patent on the electric energy distribution in vehicles by means of a bidirectional dc-to-dc switching converter.

Dr. Martínez-Salamero was a Guest Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (August 1997 issue) for the special issue on Simulation, Theory, and Design of Switched-Analog Networks. He was a Distinguished Lecturer of the IEEE Circuits and Systems Society in the period 2001–2002. He is the Director of the Automatic Control and Industrial Electronic Group (GAEI), a research group on industrial electronics and automatic control whose main research fields are power conditioning for vehicles, satellites and renewable energy. He served as President of the Joint Spanish Chapter of the IEEE Power Electronics and Industrial Electronics Societies in the period 2005–2008.