

A Network of Automatic Control Web-Based Laboratories

H. Vargas, J. Sánchez, C.A. Jara, F.A. Candelas, F. Torres, and S. Dormido

Abstract—This article presents an innovative project in the context of remote experimentation applied to control engineering education. Specifically, the authors describe their experience regarding the analysis, design, development, and exploitation of web-based technologies within the scope of automatic control. This work is part of an inter-university project known as AutomatL@bs, in which seven Spanish universities joined efforts to share their experimentation resources across the Internet. The paper begins by providing a background of how the development of virtual and remote control labs with pedagogical perspectives should be addressed. In particular, we present examples of remote labs developed by two of the university groups taking part in AutomatL@bs. We then present the automatic booking system that manages the access of users to each laboratory's didactical setup. Next, we show the integration process of every component into a Learning Management System (LMS). Finally, an overall system assessment of the students' perception of the quality of the experimental environment as a learning tool is analyzed.

Index Terms—Distributed system, internet, process control, remote laboratory, distance learning.



1 INTRODUCTION

FOR more than 10 years, most educational institutions around the world have coped with the challenge of adapting traditional mechanisms of teaching/learning to the current habits of modern society, for which the Internet is becoming the main channel used to convey information regarding any subject. For instance, in the European context, this problem was addressed by introducing the educative community to the European Space for Higher Education (Bologna process), in which the Internet plays a key role in university studies [1].

The implementation of a new Internet-based education paradigm is not a problem for the humanities and social sciences. The current Internet textual/multimedia resources are sufficient to fulfill their goals. In contrast, this challenge is not an easy task when engineering and sciences studies are the scope of application [2]. In addition to textual/multimedia information and other resources needed to provide the theoretical aspects of online engineering courses, *web-based laboratories* should be included. This is particularly true for control engineering, an inherently interdisciplinary field, in which mathematics plays a fundamental role and where advances are made through a combination of math, modeling, computation, and experimentation [3], [4], [5].

Despite the aforementioned difficulties, it is possible to find many educational institutions that already include web-based teaching tools in their current engineering and sciences curricula. The Appendix at the end of this paper lists some representative examples. Most of these examples offer the possibility of working on either a simulated version of a physical process or a real device located at the universities' laboratories. However, such platforms have certain limitations that should also be enhanced [6], specifically:

1. These developments are mainly focused on solving the technical issues related to the building of web-based lab solutions and not providing specific software tools designed to meet these goals. In general, they do not take into account the programming issues that hinder control engineering teaching staff when designing and developing virtual and remote laboratories [7], [8], [9], [10], [11].
2. Most of these environments do not consider the social context of interaction and collaboration among students (and between teachers and students) in traditional hands-on laboratories [12], [13].
3. Unlike the Cyberlab and eLabs FEUP [14] (see Appendix), most of the cases are isolated experiences coming from university engineering departments that offer only a limited set of experiments (only those existing in their own labs).

For the *technical issues* (1), it is necessary to develop new software tools or add external facilities to the existing ones to facilitate the development of a software infrastructure that transforms a traditional lab into a web-based lab. Although control engineering teachers know the subjects they teach, they have not mastered all of the programming skills required to build web-based lab solutions. Therefore, this work presents some contributions that will help control engineering teachers in particular but will also aid other engineering fields. The solutions that we offer are related to

- H. Vargas is with the School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso (PUCV), Chile.
E-mail: hector.vargas@ucv.cl
- J. Sánchez and S. Dormido are with the Department of Computer Science and Automatic Control, Spanish University for Distance Education (UNED), C/ Juan del Rosal 16, 28040 Madrid, Spain.
E-mail: {jsanchez, sdormido}@dia.uned.es.
- C.A. Jara, F.A. Candelas, and F. Torres are with the Department of Physics, System Engineering and Signal Theory, University of Alicante, Carretera de San Vicente s/n, 03080 Alicante, Spain.
E-mail: {carlos.jara, francisco.candelas, fernando.torres}@ua.es.

Manuscript received 3 Mar. 2010; revised 6 July 2010; accepted 1 Sept. 2010; published online 23 Nov. 2010.

For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLT-2010-03-0020.
Digital Object Identifier 10.1109/TLT.2010.35.

the two main parts that make up any web-based laboratory: the client and the server sides. To develop the client sides, we use Easy Java Simulations (EJS), a tool specifically created for designing and developing interactive virtual labs; that is, the simulation and the graphical user interface (GUI). EJS has been successfully used and improved by many research groups to create virtual labs and the GUIs of their remote counterparts [11], [15], [16], [17]. Regarding solutions for the server side, there are many software options to program a real-time control loop (Matlab/Simulink, C++, Scicos, etc.), but our proposal is a tool for LabVIEW developers. It consists of a middleware layer, called JiL, enabling communication between EJS-created applets and LabVIEW applications by simple point and click actions without writing code. In this way, the control teacher can concentrate on developing the LabVIEW control application and publishing it on the Internet with this tool. The JiL middleware handles all of the communications issues involved with connecting the client-side EJS variables with the server-side LabVIEW variables. The control teacher does not have to program any communication code for either side; he/she simply links the variables using the mouse. More technical issues of the JiL approach can be found in [18], [19].

Web 2.0 emphasizes the *interaction and collaboration* concepts by attempting to emulate the existing relationships among users in a specific social context (2). This work goes one step further in this direction by including the use of a Learning Management System (LMS) to satisfy the demands that arise during any traditional experimentation session. This environment is an abstraction of a face-to-face laboratory where students can interact and collaborate with other classmates and instructors.

This work provides another interesting innovation in relation to other web-based labs. The work presented in this paper is not another isolated web-based lab solution (3); it constitutes a complete network of automatic control web-based laboratories developed in the framework of the **AutomatL@bs project** (<http://lab.dia.uned.es/automatlab>). In this context, several Spanish universities benefit from this initiative by sharing experimental resources and increasing the number of available laboratories for their students and instructors. Descriptions of the other web-based labs that form this network can be found in [20], [21].

In [22], a similar Internet-based network of facilities with examples in control engineering is reported. This system is designed to access and manipulate physical resources that can be located all around the world. However, there are some differences between that system and the one described in this paper. The first is the software architecture. The network in [22] is exclusively based on LabVIEW technology, and developers are free to design their own user interfaces; there are no design guidelines because there are specific end-users with common goals; that is, students of control engineering. In AutomatL@bs, each university that joins the consortium must design the client interfaces with virtual and remote versions of the control lab and maintain a similar look to the existing ones (interactivity, augmented reality, 3D views, etc.) The real-time control program must be LabVIEW or C coded, and the clients are developed in EJS. These requirements allow developers to include the communication facilities in the EJS-based clients by drag-and-drop actions without writing communication

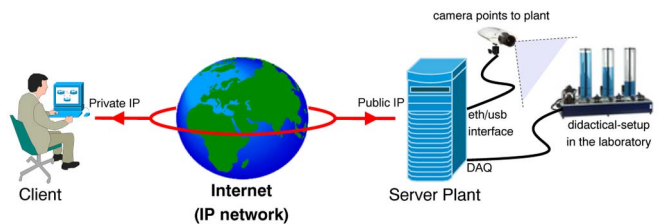


Fig. 1. Scheme of web-based lab to remotely control a didactical setup.

code. Also, the developer is not required to add TCP/IP code to the control server because that is a function of the JiL server. The second difference is the target audience. There is a total lack of teaching/learning materials and collaborative tools in [22] to support the experiments because it is mainly designed to share and lease physical setups and check algorithms for research purposes. In AutomatL@bs, a student can start working on any lab on his/her own because the materials can be downloaded. In AutomatL@bs, the user interfaces are designed to let students conduct and complete the predesigned exercises, not to modify the control code and work freely with the didactical setup. The final difference is that the end-user needs a LabVIEW kit to develop the client source code; in AutomatL@bs only a web browser is required.

This paper can also be interpreted as a survey about many of the issues that must be taken into account when beginning the development of a remote experimental environment with educational purposes. Therefore, each section of this paper provides, in sequence, the development phases that should be followed by the authors to obtain the overall infrastructure that is presented here.

The paper is organized as follows: Section 2 provides a brief background about the design and development of web-enabled control laboratories. Two sections show certain examples of remote labs used for the teaching of classic control concepts and robotics. Sections 3 and 4 describe the functional architecture of the LMS system used to support students' learning process and the booking system, respectively. Section 5 introduces the AutomatL@bs project and presents a unified full view of the experimental environment. Section 6 reports the pedagogical scenario in which the virtual and remote laboratories were used in the universities involved in the AutomatL@bs network. Section 7 describes the results of the evaluation process carried out in the framework of the AutomatL@bs project and presents some feedback from the students about the network of web-based laboratories. Finally, some conclusions about the present work and future possibilities are given in Section 8.

2 REMOTE AND VIRTUAL CONTROL LABS

2.1 Background

Fig. 1 shows the generic structure of a virtual and remote control laboratory where a remote client manipulates a real process located in the university laboratory by means of a server computer working as a middleware layer. Visual feedback from the remote equipment is also provided to the client application via a webcam that points to the equipment located in the laboratory.

In general, TCP/UDP links are commonly used for exchanging data and commands by a design pattern known

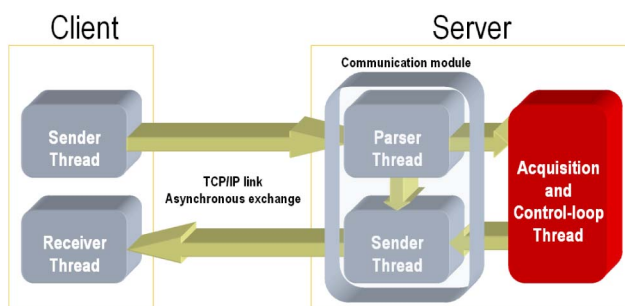


Fig. 2. Command-based architecture.

as *command-based architecture* [23]. Fig. 2 depicts the low-level scheme of this pattern.

On the server side, three concurrent tasks are commonly executed; these are the *Command Parser*, the *Sender*, and the *Acquisition and Closed-loop Control*. The Command Parser receives commands from the client, interprets them, and executes the requested actions. When no request is received, the Command Parser sleeps, leaving the processor free for other duties. Similarly, the Sender transmits to the client the measurements acquired by the control loop when directed by a command. The Acquisition and Closed-loop Control thread performs, as its name says, the data acquisition and closed-loop control of the process.

The client application implements the transmission layer needed to exchange data with the server by creating two threads, namely, *Sender* and *Receiver*. A third task, which is not explicitly shown in Fig. 2, renders the information to the final users. The graphical user interface can be a pure HTML/javascript application, or it can require a dedicated plugin such as Flash, Java or ActiveX running in a web browser [24], [25].

Developing remote laboratories with a client-server architecture is a general option, but it is not the best solution from a control engineering perspective. Applying this architecture to control experiments means that there are two information loops: the real-time control loop running locally at the server side, and the information loop running asynchronously across the network. Network delays in the information loop do not affect the real-timeliness of the control experiments, but they do delay the application of the client actions on the controlled process. In addition, the client interface does not present the results to the user in real-time because it suffers from the same communication delays. Because the processes to be controlled are not dangerous, the typical Internet delays do not produce critical situations; they only produce a delayed response by the processes. However, the use of a TCP link to exchange data avoids packet drops. The Acquisition and Closed-loop thread includes mechanisms to move the processes to a safety state when the TCP/IP channel is unexpectedly broken or a timeout error is triggered. These protections guarantee that the user will be able to restart the experiment correctly after a new connection to the remote lab is made. Currently, there is a line of research on networked control systems to develop remote labs for control engineering education in which the control loop is closed across the Internet transparently for the user [26].

Regarding the nature of the experiments included in AutomatL@bs, the system is not designed to support control engineering experiments alone. It is possible to

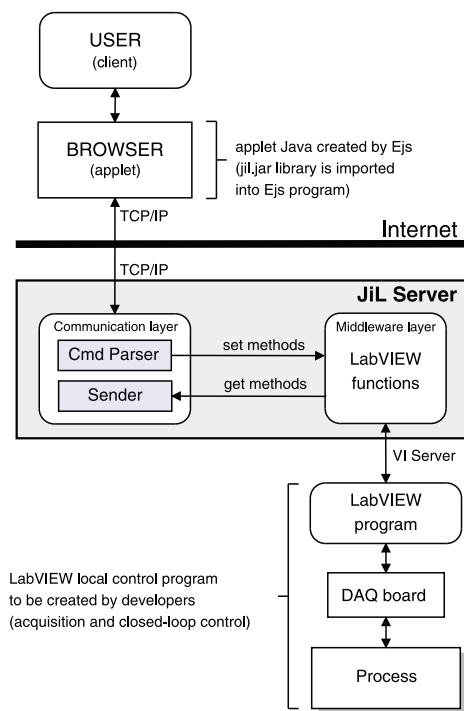


Fig. 3. Modular scheme of the JiL server approach.

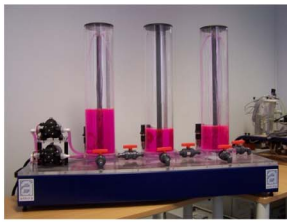
include any kind of experiment from various science and engineering fields. For example, the AutomatL@bs approach has been used for physics applications to provide web-based experiments for optics, mechanics, electricity, nuclear physics, and so on. [27]. The only difference between the architecture of these web-based labs is that now there is not a real-time control loop on the server-side.

2.2 Remote Control Lab Prototypes of the UNED

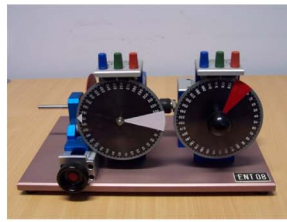
A new approach to simplify the creation of web-based labs has been developed at the Department of Computer Science and Automatic Control in the Spanish University for Distance Education (UNED). The framework relies on two software tools that are especially useful for developing these kinds of applications: EJS [28] and LabVIEW [29]. The communication issues found in this approach are resolved by the creation of generic communication modules in both the client and the server sides (see Fig. 3).

On the client side, a Java package called *jil.jar* was created to provide a generic communication interface. This package hides the TCP protocol from users, and its classes and methods set up the connections. This package can be easily integrated into EJS programs to dialog with the server. Similarly, a LabVIEW executable program called *JiL Server* operates on the server side as a middleware layer between the client and the plant. Therefore, developers are only required to create a LabVIEW local control program (known usually as VI) that only performs the closed-loop control of the plant (lower part in Fig. 3). The JiL server is responsible for all of the communication tasks in a way that is transparent to the developer.

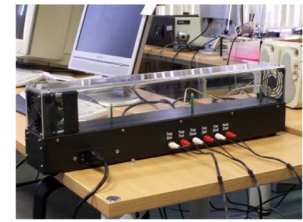
Some examples of virtual and remote control laboratories developed by the described approach are presented in Fig. 4. Figs. 4a and 4d show the three-tank system, a MIMO (Multiple Input-Multiple Output) system where one can complete liquid level control experiments [30].



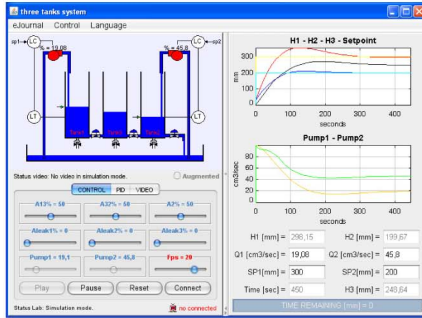
(a)



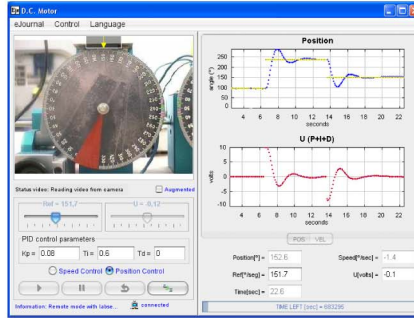
(b)



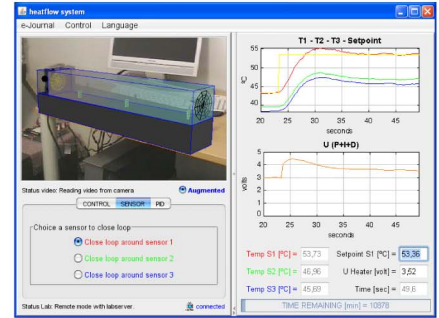
(c)



(d)



(e)



(f)

Fig. 4. Virtual and remote control laboratories of the Automatic Control Laboratory at UNED. (a) Three-tank setup. (b) DC Motor setup. (c) Heatflow setup. (d) GUI working on simulation mode. (e) GUI working on remote mode with video feedback. (f) GUI working on remote mode with augmented reality feature enabled.

Multivariable control concepts can be studied and put into practice using this laboratory module. Figs. 4b and 4e show the DC Motor, a SISO (Single Input-Single Output) system that allows one to study the dynamic behavior of the speed and position of a motor fed by a direct current source. Finally, Figs. 4c and 4f present the heatflow system that allows students to conduct practical experiments on temperature control systems with transport delays [31]. Detailed information on the modeling stage of these three processes can be found in [18], [32], [33], [34].

Every laboratory has two working modes: *virtual* (based on a mathematical model of the process) and *remote* (by accessing the real plant directly through the Internet). The GUIs are divided into two parts. The left part contains a graphical representation of the plant and a control panel to manage the system parameters. The virtual representation was developed by copying the actual didactical setup. Therefore, any change of the system state during the simulation will be automatically represented in the virtual scheme. On the other hand, when a user works on remote mode, the virtual representation is replaced by video images sent from the server. In this working mode, the augmented reality option can be enabled (Fig. 4f). With this configuration, the virtual representation of the process can be overlapped onto the video image.

The applications are found on the Internet as Java applets (developed with EJS) that are embedded in webpages. These applets are hosted and published by a web server located at UNED.

2.3 Remote Robotics Lab Prototype of the UA

The system located at the University of Alicante (UA) was entirely developed for the teaching and learning of automation and robotics. Fig. 5 shows the actual plant, which was built and assembled by a research group at UA. The main hardware components of the system are an industrial robot Scorbot ER-IX (Intelitek) with five degrees

of freedom, a small warehouse for pieces, a conveyor belt, and a rolling table. The dynamics and kinematics models of the robot were obtained from [35] and [36].

The design of the virtual and remote laboratory of the industrial robot arm is based on a supervised control architecture [37]. The surveillance and the distance programming of the robot is performed at a level of abstraction higher than simple manipulation in addition to not including the user in the control loop. Fig. 6 shows how this communication framework is implemented inside the remote system. The client sends high-level orders or trajectories to the robotic system. These trajectories have been previously executed and tested in a virtual environment, which is an exact model of the real plant. Subsequently, the client visualizes the results obtained during remote operation by means of the Reading Thread. This process is responsible for achieving current joint values of the actual robot during teleoperation.

As can be seen in the proposed scheme, the communication between the client and the server is done using the



Fig. 5. The experimental setup in a laboratory of the UA.

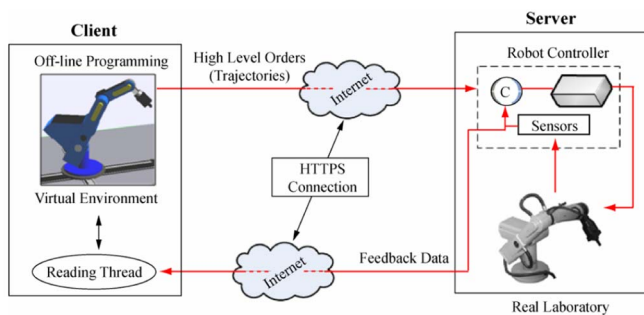


Fig. 6. Supervised control architecture implemented using the HTTPS protocol.

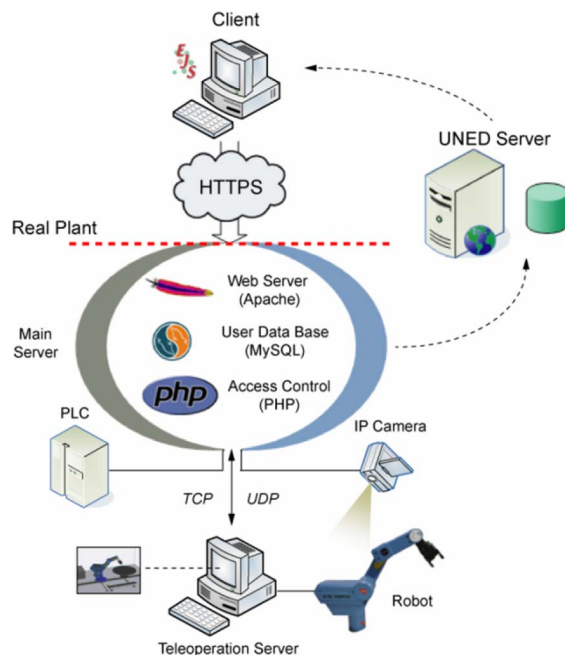
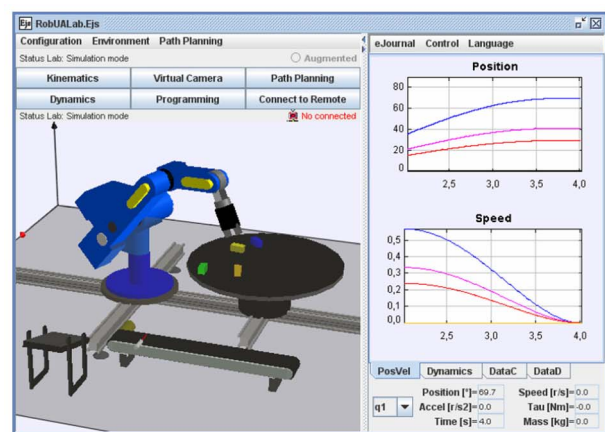


Fig. 7. Hardware and software architecture of the virtual and remote laboratory of the UA.

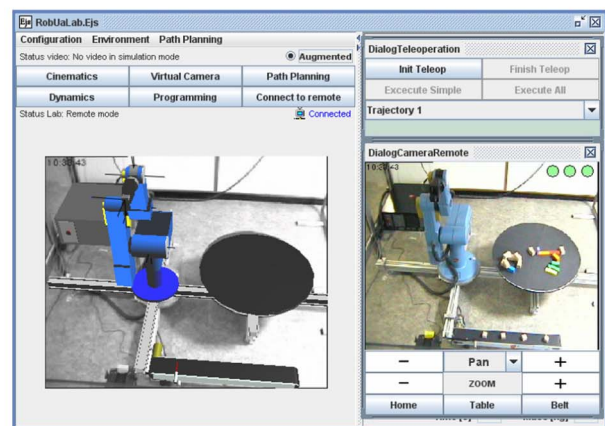
HTTPS protocol. This protocol, based on the request-response paradigm, allows authors to simplify the configuration and programming tasks for both the client and server sites. Access to the remote system is similar to that of a webpage, which prevents problems with firewalls. Moreover, all of the data sent are encrypted using the Secure Socket Layer (SSL) protocol, which provides users with secure connections and guarantees data integrity.

Fig. 7 shows the hardware and software architecture of the virtual and remote laboratory. The client is a Java applet developed with EJS that users can download from the UNED server of the AutomatL@bs project (see Section 5). In this application, the main parts are the robot model, which manages the 3D simulation, and the functions used for teleoperation.

The Main Server is the most important component in the laboratory. This computer is in charge of client communication, web administration/management of the system, and access control for the teleoperation of the real plant. The database in this computer manages reservations to use the actual robot and is synchronized with the UNED server of the AutomatL@bs project similarly to the operations of other virtual and remote laboratories.



(a)



(b)

Fig. 8. Graphical user interface of the Robotics Lab. (a) GUI of the robotics laboratory in simulation mode. (b) GUI of the robotics laboratory in remote operation mode, with augmented reality and video feedback.

Data exchanged between the client and the Main Server is codified as URL strings sent by HTTPS. These data include information such as user login, configuration parameters, and commands to be executed by the robot arm. After the client connects to the Main Server and the login authentication process is successful, these high-level orders are sent to the Teleoperation Server through a communication channel that uses TCP and UDP protocols. The computer validates these commands with a simulation based on the same robot model as the client application. This guarantees the correct use of the real plant. Finally, commands are translated into the appropriate robot language and sent to the robot controller through a serial port.

In addition, the system also includes protocols and software functions that allow users to control the power of the physical devices connected to a PLC in a laboratory. In this way, authorized users can switch on/off the lighting and the robot controller from the user interface offered by the client applet.

Fig. 8a shows the graphical user interface (GUI) of the client applet, which contains a 3D simulation of the real robot arm. This interface was developed using EJS and its 3D features. This tool includes a set of graphical elements that allow for the development of a very realistic model of the remote system. The virtual environment allows users to have a complete simulation of all of the functions of a real

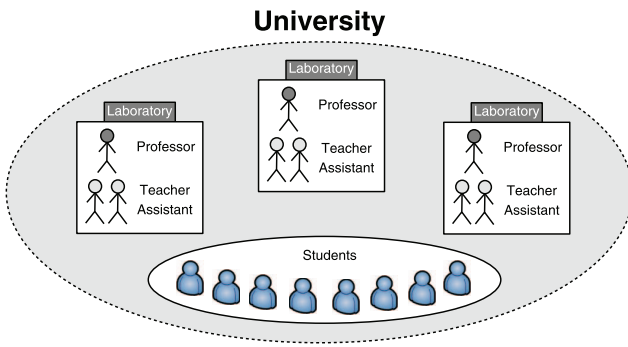


Fig. 9. Abstraction of the eMersion design.

robot arm, such as kinematics, dynamics, programming, and task planning.

The application has two options for feedback information from the real system when it is working in remote operation mode (see Fig. 8b). On the one hand, it is possible to see the video stream from an IP camera placed in the real laboratory. This option allows users to view the current state of a remote task. On the other hand, in remote operation mode, the 3D graphical representation of the virtual robot is updated with the current state of the real robot, which is directly transmitted from the robot controller. This second feature allows users to see the correlation between the virtual model and the real system during a remote experiment.

The user interface for this laboratory also provides the option of augmented reality visualization. When this feature is enabled, images from the video stream of the real system are overlaid with a 3D view of the virtual environment, with an automatic 3D perspective correction (see the left side of Fig. 8b). Moreover, this overlying is dynamic, and it is possible to change the perspective of the camera (pan, tilt, and zoom) during the teleoperation process. The augmented reality feature provides better information about the operation of the real robot than the virtual representation, and it is also a factor that encourages users to experiment with the system.

3 LEARNING AND INTERACTION SUPPORT

As shown in the previous section, the potential of web-based applications for student experimentation is obvious. However, in an e-learning scenario, students must carry out practical activities without teacher assistance. Therefore, apart from virtual and remote laboratories, additional tools are necessary for the following requirements:

1. Providing the needed documentation that allows students to autonomously carry out the practical exercises requested by their teachers.
2. Providing a common workspace where students and teachers can interact and communicate in a synchronous or asynchronous way like in a traditional laboratory.
3. Providing the possibility of creating workgroups where the components can interact like those in a face-to-face classroom.

To solve these problems, we studied the use of additional web resources to complement and assist virtual and remote

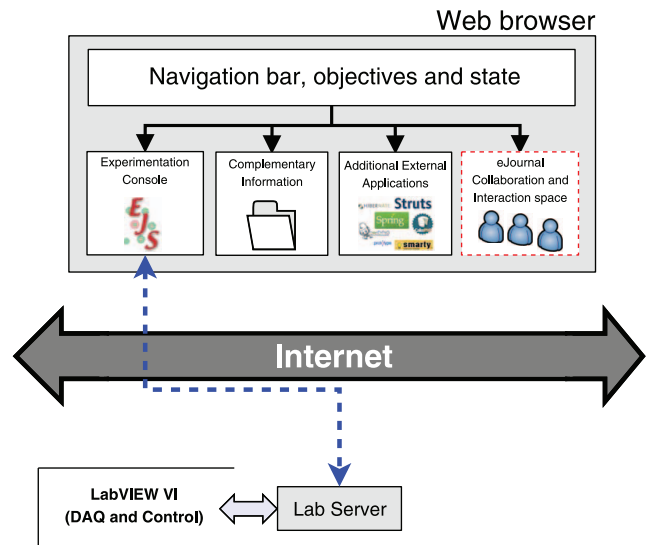


Fig. 10. Functional architecture of eMersion and web components of the experimentation environment.

laboratories. Thus, with the aim of developing a suitable learning model for students that includes the mentioned requirements, an LMS tool called eMersion [38] was chosen to support the demanded features. eMersion is a web-based application that implements a social learning model specifically designed for student education in technical subjects such as automatic control.

A developed system architecture contains certain hierarchical properties that can be found in any university. In this context, eMersion was designed taking into account three essential ideas:

1. A university has several laboratories.
2. Each laboratory is supervised by a professor, in addition to lab tutors (teacher assistants).
3. Students attend the courses given by the professors or the teacher assistants.

Fig. 9 represents this conceptual architecture of eMersion, which was initially considered as a web-based application. In this sense, a collection of independent web modules compose its environment. They cover all of the requirements for students' interaction and collaboration and support the learning process.

Fig. 10 shows the functional architecture of eMersion, which emphasizes each of the web resources that comprise the proposed experimentation environment. Next, we report a brief description of the functionality of each web resource.

- **Navigation bar.** From this web resource, students can access the rest of the options contained in the environment. Its functions are to provide information regarding the practical experiment, to inform the user about the current state of the exercise, and to give information about how to use the system.
- **Experimentation console.** This option contains a Java applet like the ones presented in Section 2. Students can carry out the practical activities requested by teachers using this web resource. If students are working on remote mode, the experimentation console will connect with a remote plant to complete the corresponding experiments.

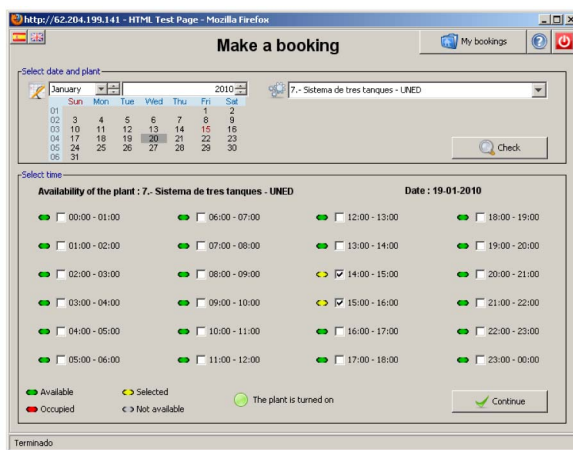


Fig. 11. Client interface of the booking system.

- **Online information.** This provides all of the theoretical and practical content required to complete an experimental session at a remote plant. The documentation must be structured clearly so it can be easily followed by students.
- **Integration of external applications.** eMersion allows developers to include additional external applications as new resources in the environment. An example of this is the complete integration of the booking system developed by the authors (see next section).
- **eJournal.** This resource plays a key role in the development of the experimental environment. It provides interaction and collaboration functionalities for group work.

4 CONTROLLING ACCESS TO THE LABORATORIES

A web-based automatic booking system was implemented to organize students' ability to access to the remote experimental resources of the environment. A student can book the plant for exclusive use on a specific date and time.

The automatic booking system is composed of two parts: 1) the booking system interface (client side) and 2) the web server, which manages and maintains the students' reservations (server side).

4.1 Booking System Client Interface

The process of making a booking involves, first, the student's request for a reservation, and second, the response of the booking system with a date and time to exclusively access a remote plant. Because the experiments were designed to be conducted online, it is not possible to book time-slots to run experiments in batch mode at night. Fig. 11 shows the user interface whereby students make their reservations for a laboratory. A green indicator at the bottom of the client interface informs users that the process is working properly and that booking is permitted. When the indicator is red, that means the process has broken down or is undergoing maintenance, and that bookings are unavailable. The system does not allow overbooking because only one student can reserve a given time-slot. Once a student books a time-slot, the time-slot designation changes to "not available" in the client interface of the booking system.

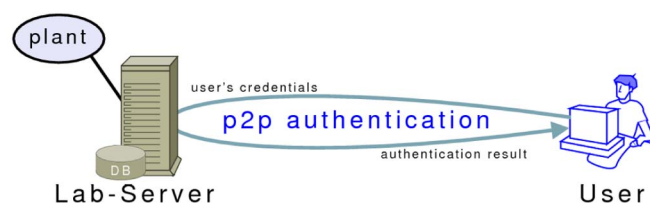


Fig. 12. Point-to-point authentication protocol.

4.2 The Server Configuration

An administrator of the booking system is in charge of both setting up the server and subscribing other new remote plants. Through the server GUI, the administrator can configure its parameters and send messages to users about broken-down processes or maintenance jobs. This server, located at the Department of Computer Science and Automatic Control of UNED, is responsible for storing all of the reservations made in the system. Therefore, this computer must be connected via the Internet to the different lab servers that provide remote experimentation services. The most relevant configuration parameters of the booking server are the following:

- Configuration of the supervisor's e-mail account for each laboratory. The system confirms the reservations carried out by users who send an e-mail to the supervisor.
- IP address of the computers that control each of the remote laboratories of the network.
- Scheduling restrictions of each experiment. Currently, the default configuration is a maximum of six hours of connection per exercise. However, users can be connected to the system only two hours per day and four per week. If a student does not make a reservation, the time is discounted from the total available.

Once a booking is made, the user can access the system only between the start-time and end-time of the reserved time-slot. Fig. 12 depicts the general idea of the implemented authentication scheme. The protocol includes the sending of a user's credentials from the client to the lab server. The server provides a simple authentication service that checks the credentials in a local database that contains a list of previously authorized users. Then, the server returns the result of the authentication to the user granting her/him access to the plant depending on the result of the check.

5 THE AUTOMATL@BS PROJECT

One of the most important results during the past years of research has been the integration of remote labs from several Spanish academic institutions into the framework of the AutomatL@bs project [39]. Several universities benefit from this initiative by sharing experimental resources and increasing the number of laboratories available for their students (the examples described in Section 2 are part of this experience). Fig. 13 shows the website's ability to access the experimental environment of the AutomatL@bs project.

The website provides information about the whole system, the available plants, the virtual and remote laboratories of the network, and so on. The purpose of this

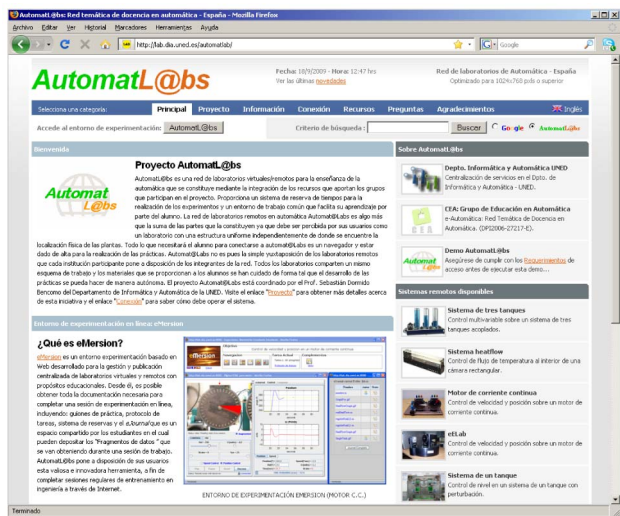


Fig. 13. AutomatL@bs project homepage.

information is to demonstrate and introduce students to virtual and remote experimentation before they access the system and begin working on the web-based labs. At this time, there is not a complete English version of the website; only the user interfaces of the virtual and remote labs are in English and Spanish. Including other languages is not difficult as eMersion and EJS have multilingual facilities.

Once the student has reviewed all of the information contained on the website and determined whether it meets the hardware and software requirements, he/she can access the experimentation environment. The network of virtual and remote laboratories of the AutomatL@bs project uses eMersion to provide a collaborative environment that allows students to share their experimental results.

Fig. 14 shows the visual aspect of the experimentation environment eMersion in remote mode during a practical session with the three-tank system. As described in Section 3, the environment is composed of five independent web applications: a navigation bar, online information (documentation), experimentation console, eJournal, and external applications.

The navigation bar gives the user access to other web resources in the environment. From the *Access Protocol* link, users can get a complete user's guide for the tool.

A collection of HTML pages accessible from the same environment provides the documentation (see lower left side in Fig. 14) needed to carry out the activities required by the teaching staff autonomously.

By using the experimentation console (Java applet), students can perform the experimental activities described in the documentation. They can save data (parameters, measurements) from their experiments in the eJournal space for later analysis.

The eJournal resource provides a shared workspace that allows users to communicate and collaborate during the learning process. Students can share their experimental results and documents with other students and/or their teacher assistants by using the different options that this tool provides (see Fig. 15). The environment does not include any software mechanisms to prevent plagiarism among students. The tutor is the only mechanism to detect copying by checking each student's written report. One improvement that could be made to prevent plagiarism would be to certify the figures and result files produced by each student with time and user information in the Ejs-based clients; in this way, the tutor could know that there are reports from different students with the same figures and result files. Finally, eMersion allows for the integration

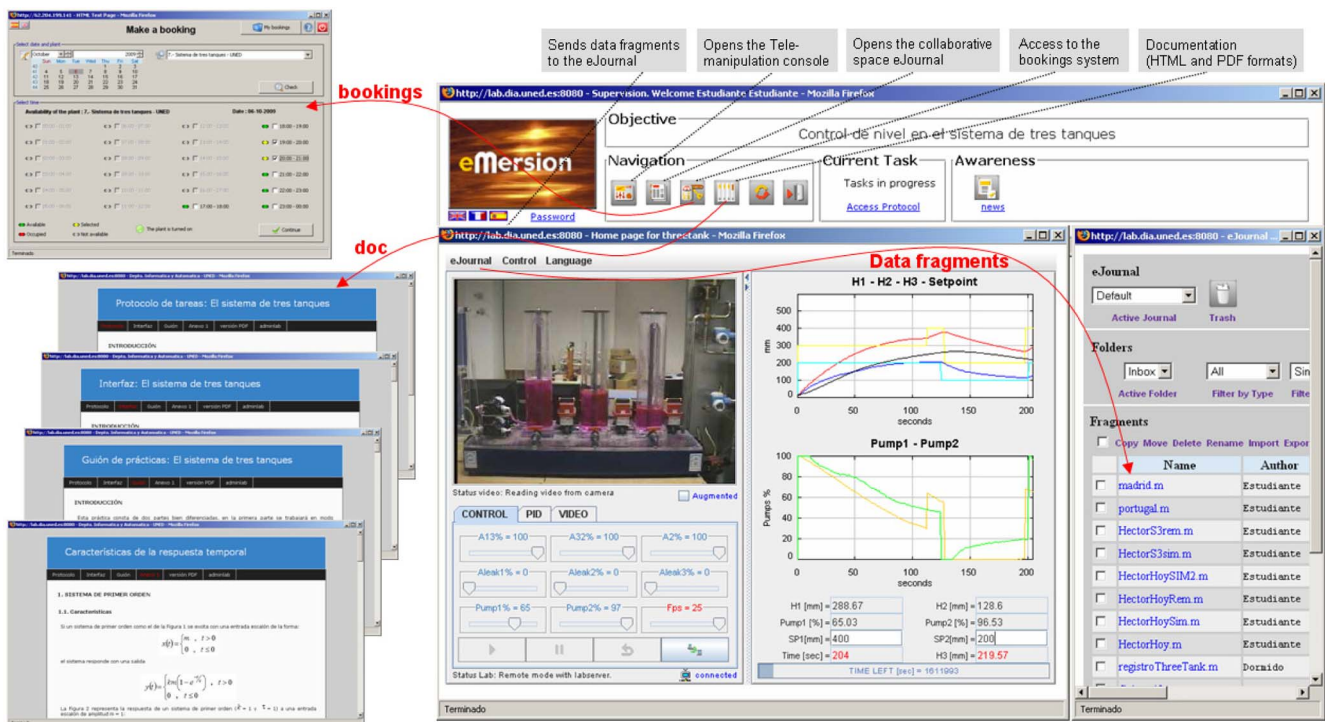


Fig. 14. Web-based experimentation environment during an exercise in remote mode with the three-tank system.

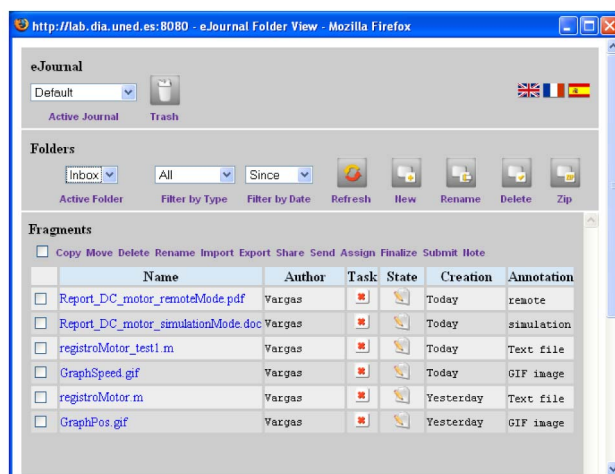


Fig. 15. The shared workspace: the *eJournal*.

of other external web applications. The automatic booking system, which was described in a previous section, is one example (see upper left side in Fig. 14). The client-user interface of the booking system has been completely integrated into eMersion. Thus, students can make their reservations from the same experimental environment in which they complete their remote experiments.

6 PEDAGOGICAL SCENARIO

The AutomatL@bs network is designed to allow students to access *any resource* from *anywhere* at *any time*, but there are pedagogical limitations. The ability to access the resource at *any time* is limited for two reasons: to teach students that they must plan and schedule their work, and to maintain some similarities with face-to-face labs. It is possible to access the labs *anywhere* because the only requirement is a Java-compatible web browser. However, a student cannot access all resources because there are restrictions. Every university group in AutomatL@bs teaches control engineering courses, but each uses different syllabuses and experiments. So, each university decides which experiments will be accessible to its students during the academic year.

Students from each university worked on three of the nine available remote systems (three from UNED and six from the other universities) offered by the AutomatL@bs project (one lab from their university and two labs from other locations). Unlike UNED, the teaching methodology of the other academic centers focused on face-to-face, hands-on activities in the presence of a teacher assistant. For this reason, some live demonstrations on the use of the remote experimentation interface during classroom sessions were done by teaching assistants. Then, students were required to practice using the system until they could operate the interfaces fluently. After several sessions, they were able complement their work with the AutomatL@bs web-based experimentation system. UNED students were told to begin working with the processes at the UNED lab in the traditional way, and, later on, they could complete their work remotely through the Internet.

During these experimental sessions, students could save their data measurements and parameters, which they used

later to write their final reports. Students then placed their reports in the eJournal space for evaluation. Teaching assistants from each university were in charge of evaluation.

The available prototypes of the AutomatL@bs project have also been fully documented. The guides have a homogeneous structure, and the content was written considering certain practical aspects from a pedagogical point of view. In general, the documentation defines a set of tasks or activities that a student should carry out so he/she can be evaluated effectively by the teaching staff. This sequence of activities was divided into two phases: PRE-Labs activities and Labs activities. PRE-Labs are based on the use of the experimentation console in simulation mode. Thus, the teaching team is assured that each student has prior knowledge of the system before using it for actual experimentation. Labs tasks are based on the use of the experimentation console in remote mode. Remote access to the system is allowed by the teaching staff once students' work in simulation mode is evaluated as satisfactory.

7 USERS' PERCEPTION STUDY

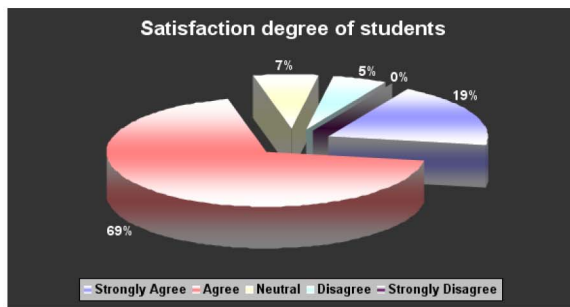
Capturing students' perception of their learning experiences is an important way to assess the quality of the experimental environment as a teaching tool. To evaluate the experimental environment, the system was tested by a total of 120 students from the seven universities that participate in the AutomatL@bs project. The main aims of this study were 1) to enable students to access other experiments that were not available at their universities, and 2) to increase the quality and robustness of the network of the virtual and remote laboratories for a large number of students and teachers with different learning and teaching concerns. Students were required to fill out evaluation questionnaires about their experiences with the system. From [40] and [41], we found some guidelines that we used to formulate questions that not only took infrastructure issues into consideration but also paid attention to the educational value of the system. Questions in which students were required to evaluate the quality of the web-based labs from a technical point of view were designed, but there were also questions about how they felt using this new methodology, so we could gather information about the educational value of the system. Some of the more relevant questions are detailed below.

Technical questions:

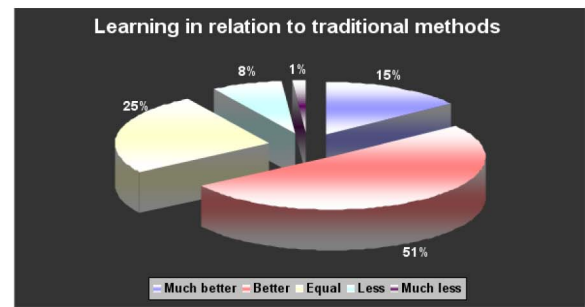
- How do you qualify the quality of the virtual laboratories?
- How do you qualify the quality of the remote laboratories?
- Have you experienced hardware or software problems?
- Did you appreciate the uniform structure of the laboratories' client interfaces?
- How was the navigation experience across the global system options?

Educational value questions:

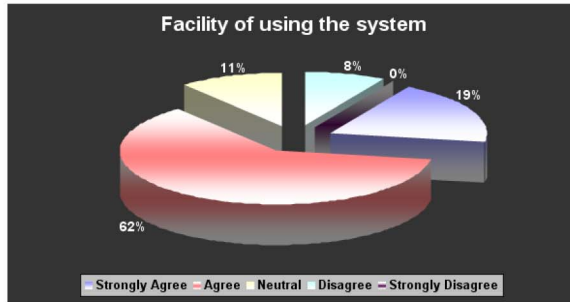
- How do you evaluate the quality of your learning using the new methodology based on web-based laboratories compared with traditional methods?



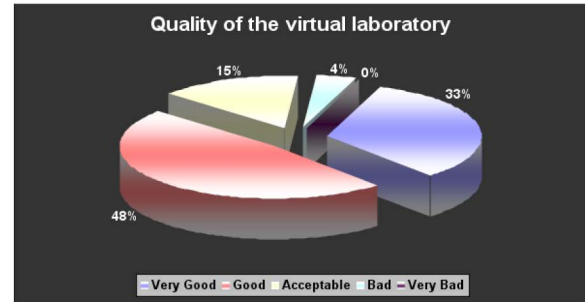
(a)



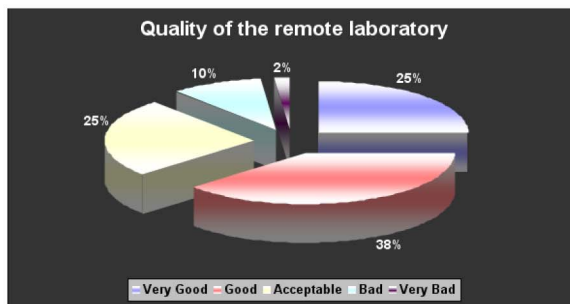
(b)



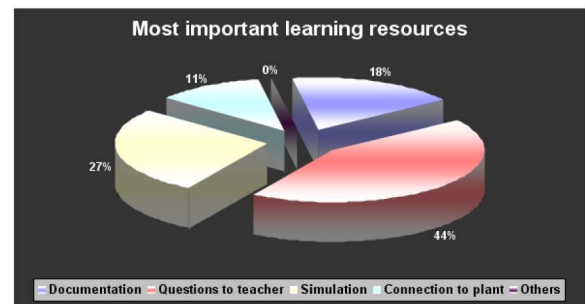
(c)



(d)



(e)



(f)

Fig. 16. Collection of pie charts summarizing the assessment results of the system.

- How do you qualify your learning speed using remote and virtual labs compared with traditional methods?
- In general terms, are you satisfied with the usability of the system used for practical purposes?
- What were the most important learning resources when you were learning to use the system?
- How would you evaluate the level of difficulty?

Although this assessment was not an exhaustive evaluation, it provided initial information regarding what is necessary or unnecessary to include in this methodology for future engineering courses.

We now discuss a collection of pie charts summarizing the results obtained in the evaluation process. Fig. 16a shows a first general view concerning whether students felt satisfied with their practical experiences. The results showed that 19 percent of students answered that they strongly agree, and 69 percent agree that they were satisfied with the system. Other questions about the advantages of using remote experiments in the educational process were also reported. The results showed that the use of new technologies, especially the Internet, encouraged students to conduct most of their practical exercises using this resource.

Figs. 16b and 16c show comparative information about learning using the new technological methods compared to the traditional methods. In most cases, dissatisfaction (around 9 percent) was found because students did not have the opportunity to work directly with the actual equipment. A way to solve this problem could be to apply an educational methodology based on “blended learning;” that is, to first carry out face-to-face classes where students can interact and experiment *in situ* with the real plant, and, afterwards, to allow them access to the experimental environment to remotely complete their practical exercises.

Regarding the quality of the virtual and remote laboratories (see Figs. 16d and 16e), most students positively evaluated their development in terms of user functionality. Any negative results may be a consequence of the quality of the Internet connection speed because slow speed will cause delays. Some of the students did their experiments using old dial-up links (56 kbps), and the exchange of data with some processes was not fast enough, causing the user interfaces to update slowly. The experiments were also tested with low-speed ADSL lines (512 kbps/128 kbps), and the results were satisfactory. Finally, Fig. 16f shows how the queries of the teaching staff and the documentation of the

practical exercises are essential resources for positive student performance.

8 CONCLUSIONS

In this paper, we presented some web-based applications for remote student experimentation in the field of automatic control and systems engineering; these include the control of a *three-tank system*, a *heat-flow system*, a *DC motor*, and the programming of an *industrial robot arm*. We described how these were developed and how they were published on the Internet. A common link was the software tool used to create the graphical user interfaces: EJS, an open-source tool that is useful for designing and developing web-based interactive simulations.

The AutomatL@bs project presented in Section 5 is a recent phenomenon. It has demonstrated its usefulness during the past three academic years in the universities that participate in the program. The eMersion environment covers most of the features required to support the learning process of students when they do their work over the Internet.

Student perceptions were also analyzed during the 2008-2009 academic year. The results show a high degree of satisfaction, and they demonstrate that the approach was implemented correctly.

However, there is also room for improvement. On the one hand, it is necessary to improve the current experimental environments to increase the feeling of physical presence. To that end, it is essential to learn more about augmented and virtual reality techniques. On the other hand, it will be important to know the impact of the next web generation, Web 3.0, on remote experimentation over the Internet. If Web 1.0 allowed users to access electronic documents, and Web 2.0 permitted students and teachers to easily exchange information, Web 3.0 will improve the experimental environment by making it more intelligent, allowing an online and synchronous communication between "remote students" and "virtual teachers." Additionally, at the time this paper was written, there was no full English version of the AutomatL@bs resources. Only the user interfaces of the virtual and remote labs have been translated into English and Spanish. This is because translating all the materials from Spanish into other languages is too time-consuming a task if it is not certain that the labs network will be used in nonSpanish universities.

APPENDIX

WEB-BASED LABS: AN INTERNET SURVEY

- Automatic Control TeleLab. University of Siena. <http://www.dii.unisi.it/~control/act/home.php>
- Weblab Deusto, University of Deusto, <https://www.Weblab.deusto.es/joomla>.
- eLabs FEUP, University of Porto, <http://elabs.fe.up.pt>.
- eMersion, Swiss Federal Institute of Technology, Lausanne (EPFL), <http://emersion.epfl.ch>.
- CyberlabOrg (coming from a European project), <http://www.cyberlab.org>.
- iLabs, iCampus, the MIT-Microsoft Alliance, <http://icampus.mit.edu/iLabs>.

- NUS Internet Remote Experimentation, National University of Singapore, <http://vlab.ee.nus.edu.sg/vlab>.
- MPRL, Polytechnic Institute of New York University, <http://mechatronics.poly.edu/MPCRL>.
- LabShare, University of Technology, Sydney, <http://www.labshare.edu.au>.
- ISILab Internet Shared Instrumentation Laboratory.
- University of Genoa, <http://isilab.dibe.unige.it>.
- The Ilough-lab, A Process Control Lab in the Chemical Engineering Lab at Loughborough University, United Kingdom, <http://www.ilough-lab.com>.

ACKNOWLEDGMENTS

This work was supported in part by the Spanish Ministry of Science and Technology under project DPI 2007-61068 and the IV Regional Plan of Scientific Research and Technological Innovation (PRICIT) of the Autonomous Region of Madrid under project S-0505/DPI/0391. The authors also would like to thank these groups for their financial support of the University of Alicante through their aid to Technological and Educative Research Groups (GITE).

REFERENCES

- [1] A. Lareki, J. Martínez, and N. Amenabar, "Towards an Efficient Training of University Faculty on ICTs," *Computers and Education*, vol. 54, no. 2, pp. 491-497, 2010.
- [2] R. Williams, "Innovations 2007: World Innovations in Engineering Education and Research," *Proc. Int'l Network for Eng. Education and Research (iNEER '07)*, pp. 279-290, 2007.
- [3] W.H. Fleming, *Future Directions in Control Theory: A Mathematical Perspective*. Soc. Industrial and Applied Math., 1989.
- [4] S. Dormido, "Control Learning: Present and Future," *Ann. Rev. in Control*, vol. 28, no. 1, pp. 115-136, 2004.
- [5] K.J. Astrom, "Challenges in Control Education," *Proc. Seventh IFAC Symp. Advances in Control Education (ACE '06)*, 2006.
- [6] N. Aliane, R. Pastor, and G. Mariscal, "Limitations of Remote Laboratories in Control Engineering Education," *Int'l J. Online Eng.*, vol. 6, no. 1, pp. 31-33, 2010.
- [7] M. Casini, D. Prattichizzo, and A. Vicino, "The Automatic Control Telelab: A User-Friendly Interface for Distance Learning," *IEEE Trans. Education*, vol. 46, no. 2, pp. 252-257, May 2003.
- [8] M. Casini, D. Prattichizzo, and A. Vicino, *Web-Based Control and Robotics Education*, pp. 127-151. Springer, 2007.
- [9] C. Martín, A. Urquía, and S. Dormido, *Web-Based Control and Robotics Education*, pp. 103-125. Springer, 2007.
- [10] NUS Internet Remote Experimentation, <http://vlab.ee.nus.edu.sg/vlab>, 2011.
- [11] S. Dormido, F. Esquembre, G. Farias, and J. Sánchez, "Adding Interactivity to Existing Simulink Models Using Easy Java Simulations," *Proc. 44th IEEE Conf. Decision and Control and the European Control*, 2005.
- [12] A.V. Nguyen, Y. Rekik, and D. Gillet, "Iterative Design and Evaluation of a Web-Based Experimentation Environment," *User-Centered Design of Online Learning Communities*, pp. 286-313, Idea Group Inc., 2006.
- [13] A.V. Nguyen, "Activity Theoretical Analysis and Design Model for Web-Based Experimentation," *Proc. Int'l Conf. Human-Computer Interaction*, 2007.
- [14] M.T. Restivo and M.G. Silva, "Portuguese Universities Sharing Remote Laboratories," *Int'l J. Online Eng.*, vol. 5, pp. 16-19, 2009.
- [15] J.L. Guzmán, H. Vargas, J. Sánchez, M. Berenguel, S. Dormido, and F. Rodríguez, *Distance Education Issues and Challenges*, pp. 131-167. NOVA Publishers, 2007.
- [16] C. Jara, F. Esquembre, F. Candelas, F. Torres, and S. Dormido, "New Features of Easy Java Simulations for 3D Modeling," *Proc. Eighth IFAC Symp. Advances in Control Education (ACE '09)*, 2009.
- [17] W. Christian and F. Esquembre, "Modeling Physics with Easy Java Simulations," *The Physics Teacher*, vol. 45, no. 8, pp. 475-480, 2007.

- [18] H. Vargas, J. Sánchez, N. Duro, R. Dormido, S. Dormido-Canto, G. Farias, S. Dormido, Ch. Salzmann, and D. Gillet, "A Systematic Two-Layer Approach to Develop Web-Based Experimentation Environments for Control Engineering Education," *Intelligent Automation and Soft Computing*, vol. 14, no. 4, pp. 505-524, 2008.
- [19] H. Vargas, J. Sánchez, Ch. Salzmann, F. Esquembre, D. Gillet, and S. Dormido, "Web-Enabled Remote Scientific Environments," *Computing in Science and Eng.*, vol. 11, no. 3, pp. 34-46, 2009.
- [20] R. Costa-Castelló, M. Vallés, L.M. Jiménez, L. Díaz-Guerra, A. Valera, and R. Puerto, "Integración De Dispositivos Físicos en un Laboratorio Remoto de Control Mediante Diferentes Plataformas: Labview, Matlab y C/C++," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 7, no. 1, pp. 23-34, 2010.
- [21] J.L. Guzmán, M. Domínguez, M. Berenguel, J. Fuertes, F. Rodríguez, and P. Reguera, "Entornos de Experimentación para la Enseñanza de Conceptos Básicos de Modelado y Control," *Revista Iberoamericana de Automática e Informática Industrial (RIAI '10)*, vol. 7, no. 1, pp. 11-22, 2010.
- [22] K.K. Tan, K.N. Wang, and K.C. Tan, "Internet-Based Resources Sharing and Leasing System for Control Engineering Research and Education," *Int'l J. Eng. Education*, vol. 21, no. 6, pp. 1031-1038, 2005.
- [23] Command-Based Architecture from NI LabVIEW, <http://zone.ni.com/devzone/cda/tut/p/id/3098>, 2011.
- [24] C. Salzmann and D. Gillet, "From Online Experiments to Smart Devices," *Int'l J. Online Eng.*, vol. 4, pp. 50-54, 2008.
- [25] H. Vargas, C. Salzmann, D. Gillet, and S. Dormido, "Remote Experimentation Mashup," *Proc. Eighth IFAC Symp. Advances in Control Education (ACE '09)*, 2009.
- [26] M. Guinaldo, J. Sánchez, and S. Dormido, "Diseño de un Sistema de Control Anticipativo Basado en Paquetes para Control en Red," *Proc. Ninth Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI '10)*, 2010.
- [27] L. de la Torre, J. Sánchez, and S. Dormido, "The FisL@bs Portal: A Network of Virtual and Remote Laboratories for Physics Education," *Proc. 14th Int'l Workshop Multimedia in Physics Teaching and Learning (MPTL14 '09)*, 2009.
- [28] Easy Java Simulations, <http://www.um.es/fem/ejs>, 2011.
- [29] LabVIEW, <http://www.ni.com/labview>, 2011.
- [30] Amira GmbH, <http://www.amira.de>, 2011.
- [31] Quanser Consulting, <http://www.quanser.com>, 2011.
- [32] 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 Education*, vol. 51, no. 1, pp. 35-44, Feb. 2008.
- [33] H. Vargas, G. Farias, and S. Dormido, J. Sánchez, "Web-Based Learning Resources for Automation Technicians Vocational Training: Illustrated with a Heatflow and Liquid Level Laboratory," *Proc. Seventh IFAC Symp. Advances in Control Education (ACE '06)*, 2006.
- [34] N. Duro, R. Dormido, H. Vargas, S. Dormido-Canto, J. Sánchez, G. Farias, and S. Dormido, "An Integrated Virtual and Remote Control Lab: The Three-Tank System As a Case Study," *Computing in Science and Eng.*, vol. 10, no. 4, pp. 50-59, 2008.
- [35] J. Denavit and R.S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *J. Applied Mechanism*, vol. 23, pp. 215-221, 1955.
- [36] L. Sciacivico and B. Siciliano, *Modelling and Control of Robot Manipulators*. McGraw-Hill, 1996.
- [37] K. Goldberg and R. Siegwart, *Beyond Webcams: An Introduction to Online Robots*, pp. 10-15. MIT, 2002.
- [38] D. Gillet, A.V. Nguyen, and Y. Reikik, "Collaborative Web-Based Experimentation in Flexible Engineering Education," *IEEE Trans Education*, vol. 48, no. 4, pp. 696-704, Nov. 2005.
- [39] AutomatL@bs, <http://lab.dia.uned.es/automatlab>, 2011.
- [40] J. Ma and J.V. Nickerson, "Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review," *ACM Computing Surveys*, vol. 38, no. 3, 2006.
- [41] M. Abdulwahed and Z.K. Nagy, "Applying Kolbs Experiential Learning on Laboratory Education, Case Study," *J. Eng. Education*, vol. 98, no. 3, pp. 283-294, 2009.



H. Vargas received the PhD degree in computer sciences from the Spanish University for Distance Education (UNED). Currently, he is working as an assistant professor in the School of Electrical Engineering at Pontificia Universidad Católica de Valparaíso (PUCV), Chile. His research interests include event-based control, networked control systems, and remote and virtual laboratories in control engineering.



J. Sánchez received the PhD degree in the sciences from the University for Distance Education (UNED). Currently he is working as an assistant professor in the Department of Computer Science and Automatic Control at UNED. His research interests include event-based control, networked control systems, and remote and virtual laboratories in control engineering.



C.A. Jara received the industrial engineer degree from Miguel Hernandez University of Elche. Currently, he is working as teacher assistant in the Automatics, Robotics and Computer Vision Group at the University of Alicante. His research interests include virtual and remote laboratories, collaborative learning, and industrial automation.



F.A. Candelas received the PhD degree in computer science from University of Alicante in 2001. He is full-time professor in the University of Alicante since 1999. His major researching interests are new technologies for teaching, virtual and remote laboratories for Automatics and Robotics, and industrial automation.



F. Torres received the industrial engineer and PhD degrees from Polytechnic University of Madrid (UPM). Since 1994, he has been at the Alicante University, as professor in Control, Robotics and Computer Vision. His research interests include robotics, manufacturing automation, visual servoing, morphological processing, and new technologies for teaching.



S. Dormido received the PhD degree in physics from Complutense University of Madrid. Currently, he is working as a full professor of automatic control in the Department of Computer Science and Automatic Control at University for Distance Education (UNED). His research interests include automatic control and web-based labs for distance education.