The Three-Tank System: A Remote and Virtual Control Laboratory using Easy Java Simulations

N. Duro, R. Dormido, H. Vargas, S. Dormido, J. Sánchez, R. Pastor, S. D. Canto

Abstract—**Current Internet-based networking technologies provide in engineering education the ability to enhance and replace the presence at the traditional control laboratories with a remote or simulated experimentation session. This paper presents a complete virtual and remote laboratory to make experiences on a nonlinear MIMO system: the three-tank plant. The experiment can be controlled exclusively with a standard web browser that provides the platform to transmit information as well as an environment to run Java applets. So, no additional software is needed at all. The client-side of the virtual/remote lab has been developed using the programming support provided by *Ejs*, an open-source tool to generate powerful Java applications and applets without necessity of knowing advanced programming.**

I. Introduction

QUALITY education in technical science is always connected with two levels of knowledge – theoretical and practical. While theoretical knowledge can easily be transferred to students in classical schoolrooms, engineering way of thinking and problem solving can be accepted by students only after hours of work in the traditional lab. And this point represents a serious drawback in high-level distance teaching.

Fortunately, the impressive development of Internet technologies in the last few years has contributed to increase the relevance of web-based teaching and learning in many research fields. Automatic Control is one of the technical areas in which the impact of new technologies to develop new tools for distance learning has been bigger [1]. A thorough treatment about control education by means of web technologies has been reported in [2].

Traditionally, web-based laboratories are divided in two classes according to the nature of the system: virtual and remote labs. In a virtual laboratory the simulation software of physical processes is provided [3]-[5]. In a remote web-based laboratory, real physical processes can be accessed via Internet [6]-[8].

Manuscript submitted March 6, 2005. This work was supported in part by the Spanish Ministry of Research under Grant DPI2004-01804.

N. Duro (corresponding author), R. Dormido, H. Vargas, S. Dormido, J. Sánchez, R. Pastor and S. D. Canto are with the Departamento de Informática y Automática. ETSII. UNED. 28040 Madrid. Spain ({nduro,raquel,hvargas,sdormido,jsanchez,rpastor,sebas}@dia.uned.es).

Students in control engineering education should achieve knowledge and skills of control systems modeling in order to develop controllers that enforce performance requirements. Once a controller is designed and implemented, observation of the resulting dynamic gives valuable insight into design concepts. Making use of web-based laboratories, students can observe dynamic phenomena that are often difficult to explain in written form. Furthermore, interactive experimentation on real-world plants improves the motivation of students and also develops an engineering approach to solve realistic problems. Simulation is a proper way to complement control education but in general, it cannot replace the experimentation on real plants, since a simulation is only as good as the model is, and a model is just an approximation which cannot reproduce every aspect of the process, such as, for instance, unexpected non-linearities.

In a remote web-based laboratory, students can access to the lab from a remote location that allows them to interact with real experiments. They can change control parameters, run experiments, see the results, and download data through the web [9]. Furthermore, an experimental approach to explain the most usual control system benchmarks could indeed improve the motivation of students, avoiding to be constrained just to look at simulations as results of their work.

In this context, the three-tank system has been proposed as benchmark system for different purposes, as test system for fault detection and identification, as well as for reconfigurable control [10]. It has received a great attention because it presents interesting properties in both control education and research. The system exhibits typical characteristics of constrained hybrid system [11] and has been proven useful to serve as a test-bed for algorithms related to state estimation, and control or identification of hybrid systems. For these reasons, the three-tank system is used to show the results of different control strategies and as an educational tool in teaching advanced control techniques.

The paper is organized as follows. The three-tank process is briefly described in Section II. Section III is devoted to introduce some basic concepts of Easy Java

Simulations (*Ejs* for short), a software tool that helps create dynamic, interactive scientific simulation in Java language. The implementation of the three-tank virtual lab in *Ejs* for interactive control education is described in Section IV. A brief description of the connection of the *Ejs* view with the remote real plant is described in Section V. Section VI presents two control experiences with the virtual/remote lab of the three-tank system using *Ejs*. Finally, Section VII gives some concluding remarks and considerations about further work.

II. The three-tank system

To develop the virtual lab of this work, the DTS200 three-tank system manufactured by Amira GmbH [12] has been modeled (Figure 1).

Fig. 1. The three-tank system by Amira.

The plant consists of three cylinders T1, T2 and T3 with the cross section *A*. The full structure of the plant is shown in Figure 2. These cylinders are connected serially with each other by pipes of cross section S_n. On the left side of the tank T2 is the single outflow valve, which has a circular cross section S_n. The output-flowing liquid is collected in a reservoir located under the three tanks. This reservoir supplies pumps 1 and 2 with liquid. And these pumps represent the input flows of tanks T1 and T2.

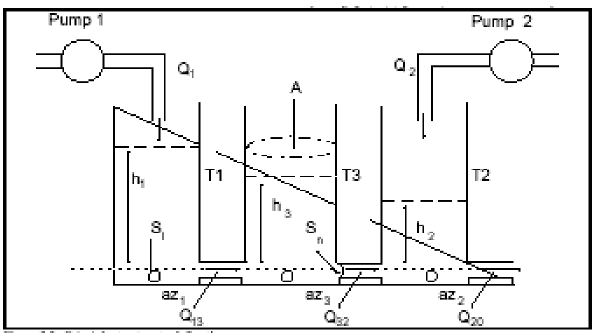

Fig. 2. Full structure of the plant.

The plant is a closed system, where the liquid that arrives to the reservoir from the tanks returns to the tanks via the pumps. However, these pumps will be switched off automatically when the liquid in T1 or T2 exceeds an upper limit. Besides the outflow valve of T2, also there are five more valves in the system. Two of them are used to connect each two consecutive tanks (one for the T1-T3 connection and another one for the T3-T2 connection), and they can be manually adjustable to close the connection between two consecutive tanks. The other three valves are placed at the bottom of every tank; these are called the leak valves. With these valves, it is possible to drain manually a tank.

In the global system, the pump flow rates correspond to the input signals of the process, and the levels of T1 and T2 are the output signals. All of them can be used for control purposes.

A. The mathematical model

The mathematical model of the plant responds to the following balance equations:

$$A\frac{dh_1}{dt} = Q_1 - Q_{13} - Q_{1leak} \quad (1)$$

$$A\frac{dh_3}{dt} = Q_{13} - Q_{32} - Q_{3leak} \quad (2)$$

$$A\frac{dh_2}{dt} = Q_2 + Q_{32} - Q_{20} - Q_{2leak} \quad (3)$$

where h_1, h_2 and h_3 represent the liquid levels in each tank, A represents the cross section of the tanks, Q_1 and Q_2 denote the flow rates of pumps 1 and 2, and Q_{ij} denotes the flow rates between tank T_i and T_j ($j = 0$ represents the output of the system). These three balance equations mean that the change of the volume in every tank is equal to the sum of the flow rates that arrive and leave the tank.

However flows Q_{13}, Q_{32} and Q_{20} are still unknown in equations (1), (2), and (3). To obtain them, the Torricelli's Law is used:

$$Q_{ij} = az_i S_n \operatorname{sgn}(h_i - h_j)\sqrt{2g|h_i - h_j|} \quad (4)$$

where az_i is the outflow coefficient, sgn(z) is the sign of the argument z, and g is the gravitation constant.

So, the resulting equations to calculate the partial flows are:

$$Q_{13} = az_1 S_n \operatorname{sgn}(h_1 - h_3)\sqrt{2g|h_1 - h_3|} \quad (5)$$

$$Q_{32} = az_3 S_n \operatorname{sgn}(h_3 - h_2)\sqrt{2g|h_3 - h_2|} \quad (6)$$

$$Q_{20} = az_2 S_n \operatorname{sgn}(h_i - h_j)\sqrt{2gh_2} \quad (7)$$

Moreover, the valves of the pipe connections, the pumps, and the leaks have been modeled using the usual relationships [13].

III. Fundaments of Easy Java Simulations

Easy Java Simulations is a freeware, open-source tool developed in Java, specifically designed to create interactive dynamic simulations [14].

Ejs was originally designed to be used by students for interactive learning under the supervision of educators with a low programming level. However, the user needs to know in detail the analytical model of the process and the design of the graphical view. *Ejs* guides the user in the process of creating interactive simulations, in a simple and practical way.

The architecture of *Ejs* derives from the model-view-control paradigm, whose philosophy is that interactive simulations must be composed of three parts: the model, the view, and the control. According to that, the steps to build an application in *Ejs* are the following: (1) To define the model is necessary to specify the variables that describe the system and the mathematical equations interrelating them; (2) define the view in order to represent the states of the process; and (3) define the control in order to describe the actions that the modeler can execute above the simulation.

These three parts are interconnected because the model affects to the view and the control actions affect to the model. Finally, the view affects to the model and to the control because the graphical interface can contain information about them.

Ejs implements its particular procedure to create specific interactive simulations since it removes the control part. Consequently, *Ejs* simulations are created by specifying a model to be run by the *Ejs*'s simulation engine and by building a view to visualize the model state and that readily responds to user interactions.

So, to define the model in *Ejs*, it is necessary to identify the variables that describe the process, to initialize them, and also to describe the mathematical equations that generate the model. The recent versions of *Ejs* support the use of Matlab/Simulink to describe and simulate the model.

The view is now the user-to-model interface of *Ejs* interactive simulations. It is intended to provide a visual representation of the model's relevant properties and dynamic behavior and also to facilitate the user's interactive actions. *Ejs* includes a set of ready-to-use visual elements. With them, the modeler can compose a sophisticated view in a simple, drag and drop way. The properties of the view elements can be linked to the model variables, producing a bi-directional flow of information between the view and the model. Any change of a model variable is automatically displayed by the view. Reciprocally, any user interaction with the view, automatically modifies the value of the corresponding model variable.

To summarize, the model is the scientific part of the simulation; yet the creation of the necessary graphical user interface (the view) is the part of the simulation that demands more knowledge of advanced programming tecniques.

Once the modeler has defined the model and the view of the interactive simulation, *Ejs* generates the Java source code of the simulation program, compiles the program, packs the resulting object files into a compressed file, and generates HTML pages containing the narrative and the simulation as an applet. The user can readily run the simulation and/or publish it on the Internet.

Easy Java Simulations, the software tool, a complete English manual for it, can be downloaded for free from Ejs' web server at [15].

IV. The three -tank system in *Ejs*

This section describes the view of the three-tank system that has been developed by *Ejs*. The model has been already described in the Section II of this paper.

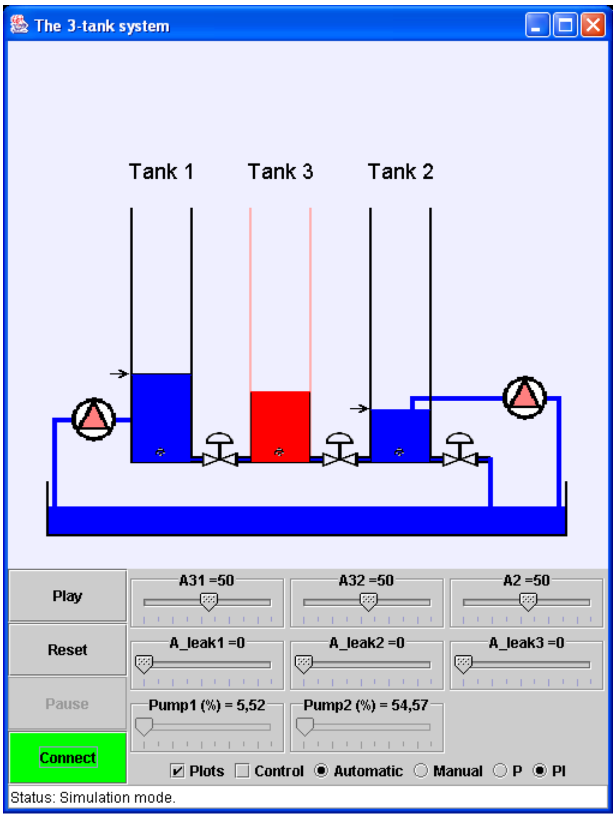

Fig. 3. View of the plant developed by using the *Ejs* graphical elements.

The view of this virtual and remote laboratory has a main window, which is presented in Figure 3. This window is composed of two parts. The upper-middle part shows a picture of the three-tank system. In the lower-middle part, it is located a panel with buttons to define different situations in the system.

The system has been represented using the following elements:

(1) *A tank*. This element has to be placed three times in the view. The tank on the left is the tank T1 of the mathematical model, the tank in the middle is the tank T3, and the tank on the right is the tank T2.

(2) *A reservoir* placed under the three tanks. This deposit collects the liquid that leaves the three-tank system, across the outflow valve located in tank T2.

(3) *A valve*. Six valves are needed in the system. Two of them are used to connect tanks T1 and T3 (in the view, the tank in the middle), and tanks T3 and T2. Another valve, which it is located on the right of tank T2, represents the outflow valve of the system. The other three valves are located at the bottom of the three tanks and represent the leaks of the tanks, which can be manipulated manually for the user.

(4) *A pump*. Two pumps are necessary in the system: one to supply liquid to the tank T1 and one more to supply liquid to the T2. They are located on the right and left of the view.

(5) *An arrow*. There are two arrows in the view that represents the set-points of the PI controllers. These set-points can be changed interactively by dragging the arrows up and down.

At the bottom of the main window, there are a set of buttons to manage the operation process of the system. On the left, there are four buttons (*Play*, *Reset*, *Pause*, and *Connect*) to specify how the system must work. The *Connect* button lets users to work in remote mode using the real three-tank system running at the university lab (this feature is described in Section V). If the user does not push the button *Connect* the application works in simulation mode. Either simulation or remote, the results are very similar because the simulation represents faithfully the behavior of the real system.

There are two radio buttons, *Manual* and *Automatic*, to select the operation mode regardless the system is operating with the simulation or with the remote plant. If the automatic mode is on, the control of the system can be done by using a P or a PI controller. To set up this control strategy, there are two additional radio buttons in the panel. In manual mode, the system works without control and the user can fix the voltage of both pumps.

To manage the opening of the valves, there are three sliders, *A31*, *A32*, and *A2*, which can range from 0% (full closing) to 100% (full opening). There are three more sliders to open and close the leak valves (*A_leak1*, *Aleak_2*, *A_leak3*, which can vary between 0% and 100%), and another two ones to fix the opening percentage of the two pumps when the system is in manual mode (*Pump1*, *Pump2*).

Finally, there are two check buttons to open two windows. By selecting *Plots*, a window with four scopes is displayed presenting the evolution of the main variables (the controlled and manipulated variables of the process); by selecting *Control,* it is shown another window with the value of the parameters of the two controllers and the control variables.

V. THE SERVER-SIDE

The architecture of the remote lab is based on a single client/server structure [6] [9], that is, the same computer performs as both the web server and controller (Figure 4). The client-side is the *Ejs* view, which can play as virtual (simulation mode) or remote lab (real system). The server-side is a computer running a HTTP server and the control loop. The web server lets students download the *Ejs* applet. The control loop governs the system according to the data packets sent from the *Ejs* view. The control parameters sent from the view to the controller are: control mode (manual/automatic), PI parameters (K_{p1}, T_{i1}, K_{p2}, T_{i2}), manual pumps settings (v_{1m}, v_{2m}), and set-points of T1 and T2 (sp_1, sp_2). The information returned to the view is composed of six values: sampling time, process manipulated inputs (v_1, v_2), and the liquid levels in tanks (h_1, h_2, h_3).

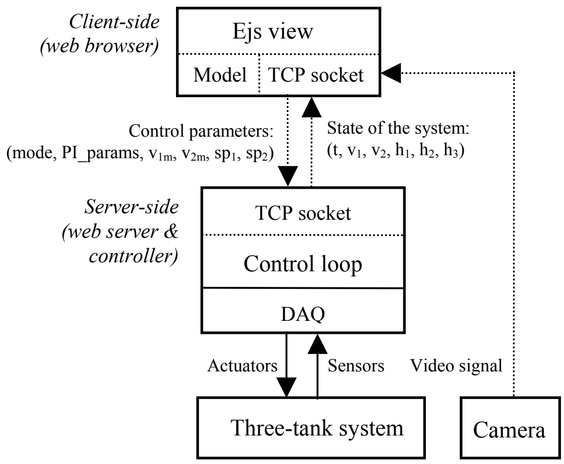

Fig. 4. Single client/server structure.

The interface of the client-side has been described in the previous section since it is the view developed by *Ejs*. This interactive view is able to play either as virtual lab by using the mathematical model, or as remote lab making a TCP connection with the controller running at the server-side. That is, a TCP channel among view and plant replaces the local channel among view and model when the interface is used as remote lab.

This switch is done when a user pushes the *Connect* button in the view. From this moment, every user action sends a data array to the server, obtaining as response a vector with the current state of the plant. The composition of the data packets is described in Figure 4. The length of the data packets is 42 bytes for the user data and 20 bytes for the system state.

The controller has been developed by the graphical programming language of LabVIEW [16]. The controller is composed of two different information loops: the asynchronous tuning loop, and the synchronous control loop. The first one is closed across the Internet with the *Ejs* view, and it suffers the unpredictable network delays. Regarding the transmission direction, the information circulating via this loop can contain the user's actions or the system's state. The control loop is running in real-time at the frequency specified for the instructor (the user will change this parameter in further versions).

At the server side, both loops exchange information by common global variables. These variables are the controllers' parameters, the set-points, the pumps settings, and the levels. In this way, every time a user changes any value in the *Ejs* view, the new value is sent via the information loop to the server-side, and it is written in its corresponding global variable; every sampling time the control loop reads this variable and applies the new value to the system.

VI. EXAMPLES OF INTERACTIVE CONTROL EXPERIENCES

This section describes two control experiences with the three-tank system by using the virtual laboratory developed with *Ejs*. The initial conditions of the system are:

1) The level of tank T1 is 250 mm, the level of T3 is 200 mm, and the level of T2 is 150 mm. These three levels can range from 0 mm to 630 mm.
2) The two connection valves between tanks T1-T3 and T2-T3 are opened at 50%.
3) The output valve, located on the right side of the second tank, is opened at 50%.
4) All the leak valves are totally closed.
5) The pump of tank T1 is working at 5.5%. The pump of T2 is working at 55%.
6) The two PI controllers have the same control parameters: $K_p = 3$ and $T_i = 1$.

The experience begins with the system working in manual mode, and it is switched to automatic mode at time 150 sec. At time 300 sec, the set-point of tank T1 is changed to 450 mm, and afterwards, at time 780 sec the set-point of the T2 is fixed to 250 mm. Both controllers work satisfactorily, as it is shown in Figure 5.

Figure 5 presents four scopes. The top row shows the evolution of the controlled variables (T1 and T2 levels) and their set-points. The bottom row in Figure 5 shows the evolution of the manipulated variables, that is, the pump's voltages. The left scope represents the behavior of the pump that supplies the flow to the tank T1, and the right one represents the pump to supply the T2. These two variables are changed due to the actions of the PI controllers.

It can be noticed that the evolution of the controlled variables is very suitable, because these variables reach the set-points very fast. As well the evolution of the manipulated variables is good since they get to a stationary state without problem, after an oscillating transitory.

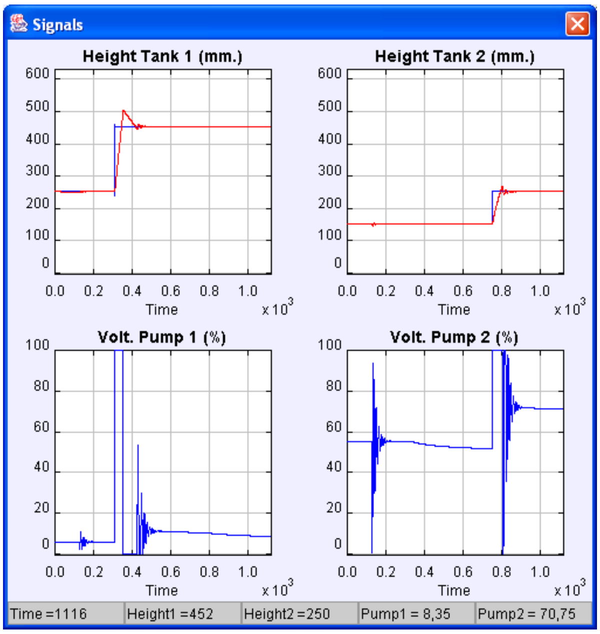

Fig. 5. Evolutions of the controlled and manipulated variables of the simulated process in the first experience.

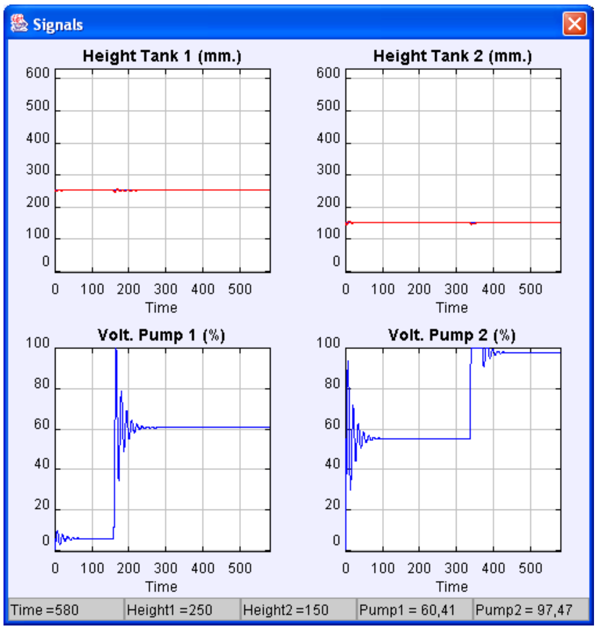

Fig. 6. Evolution of the controlled and manipulated variables of the simulated process in the second experience.

The second experience begins with the same stationary state that the previous one, but now it is tested that the system is able to cope with disturbances efficiently.

In this case, the experience is started in automatic mode and, at time 160 sec, the leak valve of tank T1 is opened from 0% to 50%. After that, a new disturbance is introduced in tank T2, opening its leak from 0% to 50% at time 330 sec.

The result of the experience is displayed in Figure 6. The beholder can conclude the results of the experiment are satisfactory. The controlled variables are practically constant and the manipulated variables have a good behavior to fulfill the requirements. So, the system is able to reject disturbances and maintain the set-point of the levels without problems. To validate the results of the virtual lab, the same experiences have been conducted in the remote lab, obtaining a similar behavior.

VII. Conclusions and Further Work

Research in remote and virtual experimentation for engineering education is a relatively young field. However, it has experimented a considerable growth in recent years, especially in higher distance education. But the procedure to transform a physical system in an interactive virtual/remote lab is not so easy yet. First, to develop the interactive views of the client-side, that is, the virtual labs, skills on advanced graphical programming are needed. In this point, *Ejs* has proved to be a very worthy solution to improve the learning in control engineering [17]-[20].

However there is some further technical work to be done regarding the creation of remote labs. The capabilities of *Ejs* have still to be improved to ease the programming of the connections with the server-side more intuitively, specially, if NI LabVIEW is chosen as software tool to create the server-side.

In relation with the web-based laboratory presented in this paper, its design allows a full replacement of the face-to-face experimentation with the physical process by a remote session using either the simulation or the real system. This lets instructors turn the traditional way of studying into a more flexible and personalized learning, reduce the compulsory presence of campus, and foster a more creative work. Experimentation on the developed laboratory provides a fundamental educational feature: students can observe on-the-fly the resulting dynamics and be aware of some physical phenomena that are tricky to explain from just a theoretical point of view.

Current and further work in this point is the building of a battery of web-based labs to offer a complete range of experiences to students (at this point in time, two and four tanks labs are going on). These virtual/remote labs must cover topics going from the basis to the advanced. Granted accesses, resource booking and security issues are being including in the infrastructure of the web-based lab.

References

[1] S.E. Poindexter, B.S. Heck. "Using the web in your courses. What can you do? What should you do?", *IEEE Control System Magazine* 19 (1), pp. 83-92 (1999).

[2] S. Dormido, "Control learning: Present and future", *Annual Reviews in Control*, 28 (1), pp. 115-136 (2004).

[3] J. Sánchez, F. Morilla, S. Dormido, J. Aranda, and P. Ruipérez. "Virtual control lab using Java and Matlab: A qualitative approach", *IEEE Control Systems Magazine*, 22 (2), pp. 8-20 (2002).

[4] D. Gillet, G. Fakas, Y. Rekik, K. Zeramdini, F. Geoffroy, and S. Ursulet. "The Cockpit: An Effective Metaphor for Remote Experimentation in Engineering Education", *International Journal of Engineering Education*, 19 (3), pp. 389-397 (2003).

[5] S. Dormido, "The role of interactivity in control learning. PlenaryLecture", *15th IFAC Symposium on Advances in Control Education*, Oulu, pp. 11-22 (2003).

[6] J. Sánchez, S. Dormido, F. Morilla, and R. Pastor. "A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum", *IEEE Transactions on Education*, 47 (3), pp. 321-329 (2004).

[7] R. Pastor, J. Sánchez, and S. Dormido. "A XML-based framework for the development of web-based laboratories focused on control systems education", *International Journal of Engineering Education*, 19 (3), pp. 445-454 (2003).

[8] A. Valera, J.L. Díez, M. Vallés, and P. Albertos. "Virtual and Remote Control Laboratory Development", *IEEE Control System Magazine*, 25(1), pp. 35-39 (2005).

[9] C.C. Ko, B.M. Chen, and J. Chen. *Creating Web-based Laboratories*. London: Springer, 2005.

[10] B. Heiming and J. Lunze. "Definition of the Three-Tank Benchmark Problem for Controller Reconfiguration". *Proceedings of the European Control Conference*, Karlsruhe, Germany (1999).

[11] D. Mignone. *Estimation and Control of Hybrid Systems*. Automatic Control Laboratory, ETH Zürich (2002).

[12] Amira GmbH' home page at http://www.amira.de.

[13] F.G. Shinskey, *Process Control Systems: Application, Design and Tuning 4th edition*, McGraw Hill, 1996.

[14] F. Esquembre, "Easy Java Simulations: A software tool to create scientific simulations in Java", Comp. Phys. Comm. 156 (2), pp 199-204 (2004).

[15] Easy Java Simulations' home page at http://fem.um.es/Ejs.

[16] J. Travis. *Internet Applications in LabVIEW*. Englewood Cliffs, NJ: Prentice-Hall, 2000.

[17] S. Dormido and F. Esquembre. "The Quadruple-Tank Process: An Interactive Tool for Control Education", *European Control Conference ECC'03*, Cambridge (UK), September 2003

[18] S. Dormido, C. Martín, R. Pastor, J. Sánchez, and F. Esquembre. "Magnetic Levitation System: A Virtual Lab in Easy Java Simulation", *American Control Conference ACC'04*, Boston (USA), July 2004

[19] J. Sánchez, S. Dormido, R. Pastor, and F. Esquembre. "Interactive learning of control concepts using Easy Java Simulations", Plenary Lecture, *IFAC Workshop Internet Based Control Education IBCE'04*, Grenoble (France), September 2004

[20] R. Pastor, J. Sánchez, and S. Dormido. "Web-based virtual lab and remote experimentation using Easy Java Simulations" (accepted), *16th IFAC World Congress*, Prague (Czech Republic), July 2005.