A Framework to Simplify the Creation of Remote Laboratories

doi:10.3991/ijoe.v6i2.1226

Calvo, E. Zulueta
University of the Basque Country (UPV/EHU), Vitoria, Spain

***Abstract*—Building remote laboratories is not a trivial issue since they are complex systems in which a great number of factors (security, QoS, integration of information of different nature, etc.) are involved. This complexity requires the use of diverse technologies that complicate the creation of the laboratories. Current work presents a framework to ease the creation of remote laboratories (both real and virtual) from a set of reusable blocks that solve most common issues (connection, student management, experiments assessment, etc.) so the designers of the experiments may concentrate on their functionality. The followed approach proposes the use of certain technologies widely used in the control engineering community such as Labview and EJS, so the creation of a new laboratory will require the creation of two applications separately that will be integrated within the framework: (1) One Labview application to acquire process information and (2) a Java applet created with EJS used as graphical interface. The proposed framework was used with a water level automatic system to show how to add new experiments to the framework.**

***Index Terms*—Remote Laboratories, Virtual Laboratories, Control Engineering Education, Teleoperation**

I. INTRODUCTION

In most engineering disciplines settling down the concepts learnt in the classroom requires that the students interact with scale models where students have to think how to apply these concepts to real plants. Typically, this objective has been achieved in hands-on laboratories. Unfortunately, for different reasons (staff availability, Faculty timetables, etc.) these laboratories tend to be underused, yielding to a scarce use of some resources that too often tend to be expensive.

On the other side, Internet based technologies have become very popular in the last decade. This fact has converted them in a very interesting alternative for developing new education resources. It is easy to find in the literature different resources that cover different parts of the learning cycle, such as web pages that use multimedia resources to visualize certain phenomena, or virtual and remote laboratories in which the students may experiment with virtual/remote plants.

Unfortunately, building remote laboratories is more complicated that setting up hands-on laboratories. Of course, it is still necessary to configure the equipment to perform the experiment, but, in addition, a remote access infrastructure must be created and properly configured ensuring certain requirements [1]:

- Remote laboratories must integrate different types of information: Process data in real time, plant configuration and operation, interaction with other students and with the instructors, multimedia information, etc.
- Certain QoS parameters must be ensured.
- The software installed on the client side must satisfy certain requirements: It should be multiplatform, easy to install, secure and, preferably, free of charge.
- Students must be able to download the data obtained during the execution of the experiments for off-line analysis.
- Concurrent access to the system by several actors (students and instructors) must be managed.
- It should facilitate the management of the users and assessment of the experiments.

These requirements make the creation of remote laboratories a complex task. Fortunately, there is a common core to most remote laboratories that includes certain characteristics and functionalities, such as connecting the physical equipment of the laboratory, providing certain level of security, managing the students, assessing the experiments, etc. This fact endorses the use of modular approaches so the designers of the experiments may build easily new laboratories from a set of reusable components. When the functionality of the laboratory is not so exigent some commercial packages like Matlab or Labview may become an interesting alternative since they provide different modules that are relatively easy to configure. Actually, it is easy to find in the literature examples of remote laboratories built with these tools [2, 3, 4, 5]. However, these packages are more aimed to accessing remotely plants than to the creation of remote laboratories for teaching purposes and consequently do not solve education specific issues such as managing the students, assessing the experiments or enabling communications among the several actors (students and instructors) involved in the experiments.

In the current paper the authors present a web-based modular framework that includes most common functionalities. The proposed framework simplifies the creation of remote laboratories and virtual laboratories. By following this approach, the designers of the experiments use tools widely used in the Control Engineering domain such as Labview [6] and EJS [7] to create pluggable components that can be easily integrated within the framework to create real or virtual remote laboratories. The core of the framework is a web application that manages the learning resources employed by different user profiles (Administrator, Instructor and Student) as well as the operations that they may carry out over the laboratories. The framework also provides the communication infrastructure to manage the different flows of information required at the execution at the operations of the remote users.

The layout of the paper is as follows: Section 2 describes the framework and some implementation issues. Next section describes the user profiles and the operations provided by the framework. Section 4 describes the procedure of connecting new experiments. Finally, section 5 describes as a matter of example, a simple laboratory based on a automatic water level system. The article finishes with some concluding remarks and the references to future work.

II. ARCHITECTURE DESCRIPTION AND IMPLEMENTATION OF THE FRAMEWORK

Remote laboratories may operate in two different ways [8]: On-line and in batch mode. Pure on-line operation requires that control algorithms are executed in the remote applications so control actions and sensor values are sent over the Internet. This approach allows changing reference and configuration parameters while one experiment is being executed, but the nature of Internet causes unpredictable delays that may affect the execution of the experiments. On the other hand, batch mode operation avoids Internet delays since both reference and configuration parameters are sent before the experiments are started whereas remote users receive the results after their execution. However, this approach reduces the interaction degree of remote users in the experiments. An alternative located in the middle of previous approaches [9] requires autonomous controllers located at the server side that are able to process reference and parameter changes issued on-line by remote applications. This is the approach followed in the proposed framework, so physical devices are managed by local controllers that are connected to the Application Server to exchange process references and configuration parameters specified by remote users and process data that are represented graphically by remote applications.

The proposed framework implements the generic architecture presented in [10] (see Figure 1), which is composed by three main entities:

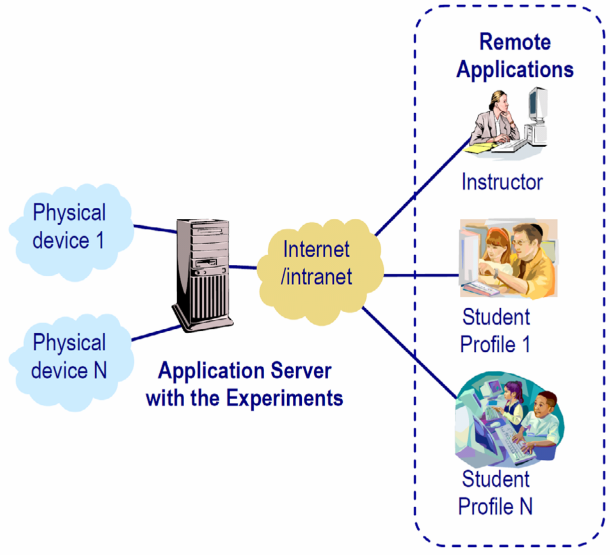


Figure 1. Generic remote access architecture

- **Remote Applications:** They provide a graphical interface executed over standard web browsers (Mozilla Firefox, Internet Explorer, etc.) which are used by the different remote user profiles (Administrator, Instructor and Student) to access remote laboratories. These applications consist of a set of dynamic web pages with certain embedded resources such as Java applets for the animations, multimedia resources for audio and video, web pages describing the available physical plants and experiments as well as other resources downloaded from the Application Server. There are different web pages per user profile since users are allowed to perform different operations that require the access to diverse resources.
- **Application Server:** It holds the web application that constitutes the core of the application. Major concerns include downloading the dynamic web pages created from the information contained in the application database, managing the sessions used by remote users to connect to the system as well as managing the process communication issues among the physical plants and remote users. In order to ease the integration of process data, the framework includes an *ad hoc Bidirectional Server* that carries out the exchange of process data and commands among remote applications and physical devices. The *Bidirectional Server* reads/writes data from/to the physical devices via TCP sockets and distributes this information among the Java applets of all interested remote users by using the Java Messaging Service (JMS) technology.
- **Process devices and local controllers:** These are the devices that take part in the remote laboratories. These devices may be real, producing measured data, or virtual, in which case data are obtained from computer simulations. Remote users may also send commands to these devices so they perform the operations required in the experiments. In addition, some experiments may require the configuration of these devices as a previous step to carrying out the experiments. In this category are also classified the local controllers that take part in the experiments.

Figure 2 shows the customization of the generic architecture which is created from a set of tools. Some of these tools have been used to create the core of the application whereas other tools are used to create new pluggable components associated to the experiments. As explained above, the core of the architecture is a web application that has been implemented with Jakarta Struts [11] which is a framework based on the MVC paradigm that simplifies the development of web applications. Thus, the creation of a new experiment requires building new JSP pages with the information contained in the application database. The framework will manage the distribution of data among the physical devices and local controllers involved in the experiments and the applets used by all connected users in the remote applications, with the so-called *Bidirectional Server*. This is an *ad hoc* component that manages the point-to-point connections between the Application Server and all physical devices / autonomous local controllers via TCP sockets at the server side and that uses JMS technology [12] to distribute the data to all remote applications.

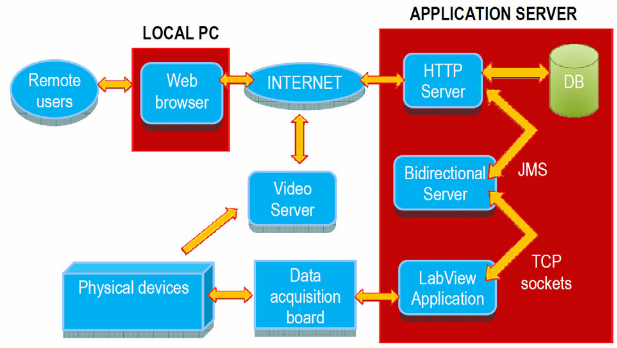


Figure 2. Custom remote access architecture

Even though different technologies could be utilized to develop the local controllers and the Java applets for the graphical interfaces of the remote applications, the authors propose as part of the framework the use of well known technologies that has been selected due to their popularity in the control domain and simplicity of use. These technologies are Labview [6] to access and control the physical devices at the server side and EJS [7] to create the Java applets used at the client side. Labview is very powerful technology to create acquisition and control applications. As Figure 2 shows, the use of TCP sockets as part of the communication infrastructure allows that Labview controllers could be embedded as part of the Application Server, or could work in stand-alone mode without altering the remote access architecture. JMS is a Message Oriented Middleware (MOM) API for sending messages between two or more clients. JMS is part of the Java Platform, Enterprise Edition, which is based on the publisher/subscriber paradigm and decouples the exchange of information between senders and receivers of information.

III. PROFILES AND OPERATIONS PROVIDED BY THE FRAMEWORK

The proposed framework distinguishes three types of remote profiles that provide separate views over the framework by grouping different operations:

- **Student:** This profile is used to execute and visualize the experiments to which students are allowed. When someone is permitted to carry out one experiment he/she will be allowed to access all didactic resources that compose it, including the Java applets that are used to interact with the plants (real or virtual). It also provides other operations such as uploading/downloading experimentation results or consulting the qualifications. The framework allows collaborative experiments in which several students assume different roles in order to achieve a global objective. In such case, the communication among students while the experiment is ongoing is compulsory, so this profile includes communication tools to exchange information among different students. As a previous step to use the laboratory, students must book the laboratory for carrying out the experiments avoiding concurrency by other remote users. There is a quota of use that defines the maximum time they are allowed to access the laboratory which will be managed by the Administrator profile.
- **Instructor:** Registered instructors will be allowed to manage both experiments and students. It allows to set up certain parameters of the laboratories so remote students have to execute the experiments with these values. Also, it allows assigning the experiments to the students. For example, users of the Instructor profile may open the student access to a real laboratory after the students have passed certain tests over a similar virtual laboratory. Instructors are permitted to add, enable or disable new resources to enlarge the laboratory or to change existing ones. Other functionalities provided by the framework to Instructors include assessing the experimentation results of the students, checking on-line the execution of the experiments and communicating on-line with the students both in a synchronous way via a chat, embedded in the framework, and asynchronously, via messages left on the platform.
- **Administrator:** They are the managers of the system. Users belonging to this profile are allowed to manage all types of users involved in the system (Administrator, Instructor and Student), including creating a new one, deleting an existing one or modifying any attribute. Also, Administrators are allowed to manage the experiments as well as defining the experiments and resources that remote students and instructors will be allowed to use. They also specify the quota of use which defines how much time students are allowed to use a laboratory to carry out one experiment. Once this quota is exceeded, students are disallowed to access the laboratory. Other operations allow checking the experimentation results of the students as well as the qualification marks obtained in the experiments. In addition, users of the Administrators group are allowed to check the access history by all remote users.

The framework allows three different modes of operation: Virtual laboratories, remote laboratories and demo mode. Using both virtual and remote laboratories in the learning process is very beneficial, especially when students use firstly virtual representations of the real plants and latter the real laboratories. As virtual laboratories are based on simulations of mathematical models they are more robust and tend to provide more flexibility. However, some characteristics of the plants may be difficult to reproduce. On the other side, pure remote laboratories provide a higher degree of reality, as students work with real plants, even though they are handled remotely. By combining both types of laboratories the students may benefit from both approaches. In addition, they would experiment the differences between the simulation and the behaviour of the real plants. The presented approach integrates both types of laboratories in the same way. Moreover, some components, in particular the graphical interfaces created with EJS, may be reused for both types of laboratories, even though in the case of remote laboratories they should be complemented with real-time audio and video streams to provide a higher degree of reality.

The framework also introduces the demo mode. This operating mode may be used for both remote and virtual laboratories. By using the demo mode, instructors may explain how a plant or device works to a group of connected students. Instructors will control the behaviour of the devices and the students are allowed to visualize it. Several communication channels such as audio and video streams as well as the chat embedded in the framework will be used to explain important issues of the experiments whereas the graphical interfaces will show the evolution of the process variables. In this working mode, students are allowed to ask questions in real-time via the embedded chat but they are not permitted to handle the plants. In the opinion of the authors, this approach is very useful to provide introductory lessons to use the laboratory or as a complementary tool to the classroom where students may visualize the behaviour of real plants as a complement to the lectures.

IV. STEPS TO CREATE A NEW EXPERIMENT

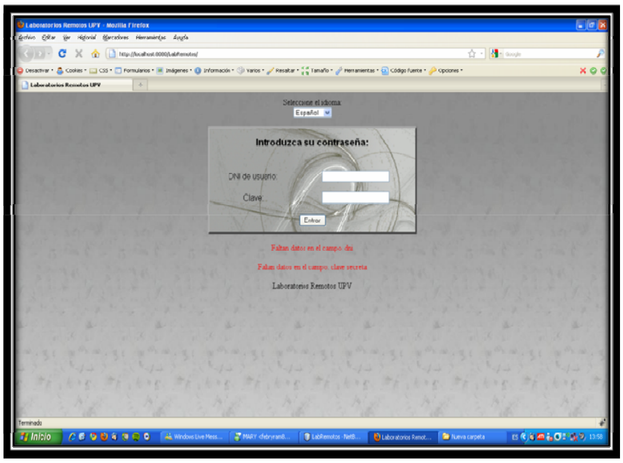
This section describes the main steps to add a new experiment to the framework. It assumes that the core web application is already installed in the Application Server. Previously, some software packages have to be installed and configured such as MySQL Server with the initial database used by the core web application and one web application server such as Sun Glassfish, which must be preloaded with the software of the Web application deployed as a ".war" file. This application is currently available in three different languages: Spanish, English and Basque.

The access to the application is granted by a web page where remote users must authenticate (see Figure 3). Automatically, users will access to the web application from where they will be allowed to carry out different operations according to their user profile (Administrator, Instructors and Students).

A. Operations carried out by the Administrator

The Administrator must create accounts for users of both Instructor and Student profiles that will be involved in the experiments (see Figure 4). Also, the Administrator must create the *Resources* that correspond to the physical devices and local controllers available at the remote laboratory. The framework uses this information to manage concurrency of use to several devices. Administrators are also responsible for connecting the physical devices to the Application Server via the *Bidirectional Server*, so they produce data that can be connected to the graphical interfaces.

Both Administrators and Instructors are allowed to create new experiments in which a set of several devices will be involved. Finally, students must be assigned to one experiment, so they may access it.


Seleccione el idioma:
Español

Introduzca su contraseña:

DNI de usuario:

Clave:

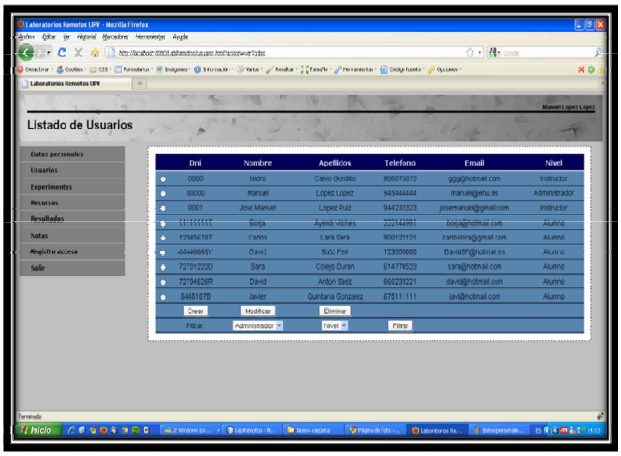
Entrar

Faltan datos en el campo: dni

Faltan datos en el campo: clave secreta

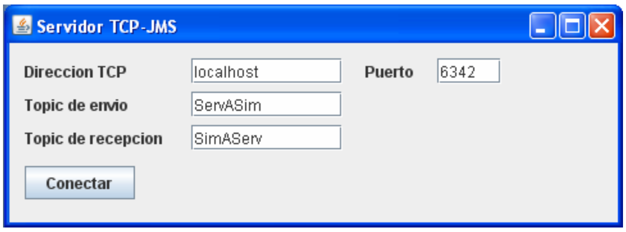
Laboratorios Remotos UPV

Figure 3. Access page to the web application


Listado de Usuarios

Dni	Nombre	Apellidos	Telefono	Email	Nivel
0000	Isidro	Calvo Gordillo	906075075	ggg@hotmail.com	Instructor
00000	Manuel	Lopez Lopez	945444444	manuel@ehu.es	Administrador
0001	Jose Manuel	Lopez Ruiz	944232323	josemanuel@gmail.com	Instructor
11111111T	Borja	Ayerdi Viches	222144991	borja@hotmail.com	Alumno
12345678T	Carlos	Lara Sera	900122121	carloslara@gmail.com	Alumno
44446666Y	David	Soto Peri	123000000	DavidSF@hotmail.es	Alumno
72731222D	Sara	Colejo Duran	614778523	sara@hotmail.com	Alumno
72734626R	David	Anton Saez	666233221	david@hotmail.com	Alumno
8448187B	Javier	Quintana Gonzalez	675111111	javi@hotmail.com	Alumno

Figure 4. Management of user accounts for remote users


Servidor TCP-JMS

Direccion TCP: localhost
Puerto: 6342
Topic de envio: ServASim
Topic de recepcion: SimAServ

Conectar

Figure 5. Configuration of the Bidirectional Server

B. Operations carried out by the Instructors

Instructors share the management of the experiments with the Administrators. They define the resources involved in one experiment as well as the students that have access to it. In addition, Instructors are responsible for creating and uploading the specific pedagogical resources related to every experiment. These resources include:

- **Web pages explaining the experiments.** These web pages will provide background information to use the remote laboratory.
- **Graphical interfaces created with EJS.** As explained, these will include the mathematical models in virtual laboratories and the communication infrastructure that sends / receives the JMS information from the Application Server.
- **Labview applications to interface physical devices.** These applications will act as local autonomous controllers of the processes. They are experiment specific and must be provided by the designers of the laboratories to the Administrator to connect it to the application.

The web pages and the EJS applets created by the Instructors will be uploaded in one folder of the system per experiment. Links to these resources will be included in the web application to be accessible by remote users. Labview applications will be configured to connect via TCP/IP to the *Bidirectional Server* (see Figure 5).

C. Operations carried out by the Students

Once student accounts have been created, the resources have been uploaded in the framework, and they have been allowed by the Administrator or the Instructors, they may start using the experiments (See Figure 6).

Students will be able to use the operations provided by the framework, including those provided specifically by the experiments as well as uploading experimentation results.

V. EXAMPLE OF USE: AN AUTOMATIC WATER LEVEL SYSTEM

The proposed framework was tested over a simple plant consisting of an automatic water level system for a prismatic tank (see Figure 7).

In this case the plant consists of a water tank in which the level must remain constant independently of the outgoing flow. The plant is supplied with level and flow transducers that measure the current level of the liquid in the tank and the incoming water respectively. The differential equation that describes the dynamics of the system is (1):

$$q_i(t) = \frac{h(t)}{R} + A\frac{dh(t)}{dt} \tag{1}$$

Where q_i denotes the input flow, h the water level in the tank, A is the section of the tank and R is the resistivity of the fluid which depends on the diameter of the outgoing pipe. By using the Laplace transform, previous equation may be represented as the following transfer function (2):

$$\frac{H(s)}{Q_i(s)} = \frac{R}{1 + A{\cdot}R{\cdot}s} \tag{2}$$

The plant is complemented with one actuator: a pump that pushes the input flow into the tank. This pump is managed by an *ad hoc* Labview application that provides the electrical signal that governs the amount of liquid that is gets into the tank according to different control algorithms. The overall block diagram of the system is represented in Figure 8.

Students may select among different control algorithms. In particular, current implementation of the Labview autonomous controller allows manual control and a PID (*Proportional Integral Derivative*) controller (3):

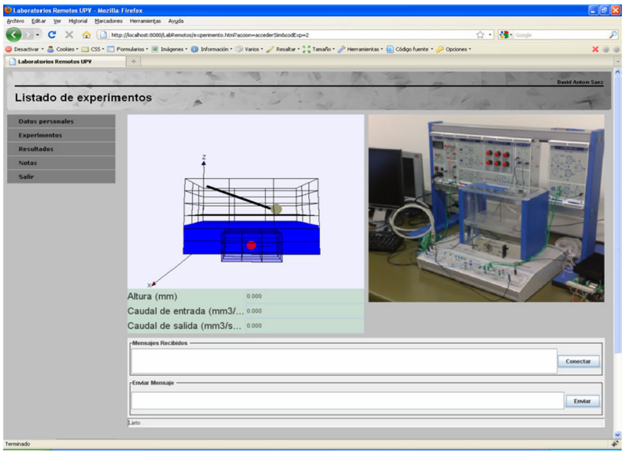
$$v_{out} = K_P e + K_D \frac{de}{dt} + K_I \int edt \tag{3}$$

Where v_{out} is the voltage applied to the pump, e is the error (difference in volts between the reference level and the current level) and K_P, K_D and K_I are the proportional, derivative and integral gains of the PID controller respectively that must be tuned.

Thus, students are required to carry out the following experiments in this laboratory:

1. In the virtual laboratory, they must tune the PID controller (3) provided in the EJS simulation to achieve certain design specifications set by the instructors (see Figure 9).
2. In the remote laboratory (see Figure 6), they must obtain experimentally the values of the transference function (2) by using the manual control.
3. In the remote laboratory they must adjust the Labview PID controller (3) to achieve the design specifications set by the instructors

Students will not be allowed to use the remote laboratory unless they have successfully tuned the PID controller in the virtual laboratory supplied by the instructors. So, the use of the real laboratory is optimized in time.



http://localhost:8080/LabRemotos/experimento.html?accion=accederSim&codExp=2

David Anton Saez

Listado de experimentos

Altura (mm) 0.000
Caudal de entrada (mm3/... 0.000
Caudal de salida (mm3/s... 0.000

Mensajes Recibidos

Enviar Mensaje

Figure 6. Using the remote laboratory

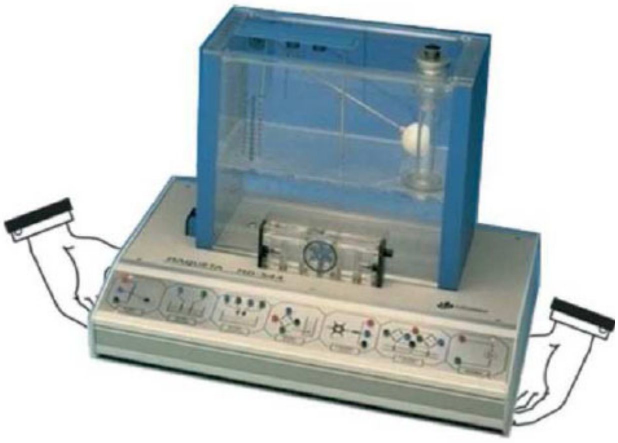


Figure 7. Tank used in the laboratory

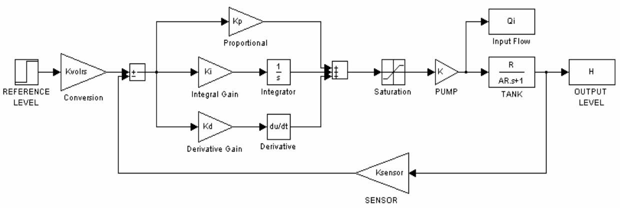
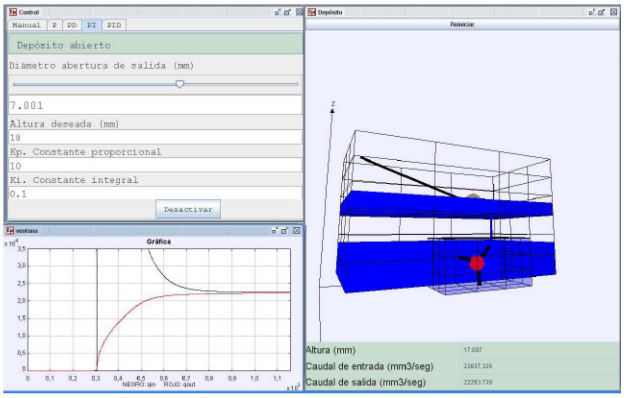


Figure 8. Block diagram of the automatic water level system



Depósito abierto

Diámetro abertura de salida (mm)
7.001

Altura deseada (mm)
18

Kp. Constante proporcional
10

Ki. Constante integral
0.1

Altura (mm) 17.097
Caudal de entrada (mm3/seg) 22837.325
Caudal de salida (mm3/seg) 22293.730

Figure 9. Tuning a PI controller in the virtual laboratory

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented a framework aimed to simplify the creation of remote laboratories. The framework is based on a set of components. One of these components is a web application that constitutes the core of the framework. This web application, which was created with Jakarta Struts, provides common operations to every user profile (Administrator, Instructor and Student). In addition, the creation of new experiments in the laboratory requires building specific components with tools commonly used in the Control Engineering domain, namely Labview to build the applications that interact with the physical devices and EJS to build the Java applets used as graphical interfaces. Thus, the creation of a new experiment requires the creation of both applications that will be connected to the core application.

The framework allows to integrate both virtual laboratories and pure remote laboratories in the same way. In fact it is recommended to use the same graphical interface for the remote laboratory and a simplified virtual version that may be provided to the students as a previous step to use the real laboratory. The article also describes the main operations provided by the framework.

In the future the authors expect to create a set of experiments that may be used in Control Engineering courses, so the developed framework may be evaluated by the instructors.

REFERENCES

[1] Dormido S. (2004) "Control Learning: Present and Future" *Annual Reviews in Control*, vol. 28 (1), pp 115-136. doi:10.1016/j.arcontrol.2003.12.002

[2] Ertugrul, N., (2000) "Towards virtual laboratories: A survey of Labview-based teaching/learning tools and future trends", *International Journal of Engineering Education*, Vol 16, pp 1-10

[3] Mergl, C. (2006) "Comparation of Remote Labs in Different Technologies", *International Journal of Online Engineering*, Vol 2, No. 4

[4] Colak, I., Demirbas, S., Sagirouglu, S., Irmak, E., (2009) "A Novel Web-Based Laboratory for DC Motor Experiments", *Computer Applications in Engineering Education*, In Press. Available online at: http://www3.interscience.wiley.com/journal/122235092/abstract

[5] Sanchéz, J.; Dormido, S.; Morilla, F. (2004) "A Java/MatLab-Based Environment for Remote Control System Laboratories: Illustrated with an Inverted Pendulum", *IEEE Transactions on Education*, vol. 47, no. 3, August, 2004 pp. 321-329. doi:10.1109/TE.2004.825525

[6] Labview. NI Website. http://www.ni.com

[7] Easy Java Simulations (EJS). Available online at: http://www.um.es/fem/Ejs/

[8] Yang, S.H., Chen, X., Alty, J.L., (2003) "Design issues and implementation of internet-based process control systems", *Control Engineering Practice*, 11, 709-720. doi:10.1016/S0967-0661(02)00181-8

[9] Dormido, S., Dormido-Canto, S., Dormido, R., Sanchez, J. (2005) "The role of interactivity in control learning", *International Journal of Engineering Education*, 21, (6), 1122-1133

[10] Calvo, I., Marcos, M., Orive D., Sarachaga I. "Building Complex Remote Learning Laboratories", *Computer Applications in Engineering Education*, Published Online**:** 22 Jan 2009. Available at: http://www3.interscience.wiley.com/journal/121649745/abstract . doi:10.1002/cae.20239

[11] Cavaness C., (2004) "*Programming Jakarta Struts*", 2nd Edition, O'Reilly Media Publishers, ISBN: 978-0-596-00651-8

[12] Java Messaging Service (JMS) Specification. Available online at: http://java.sun.com/products/jms/docs.html

AUTHORS

I. Calvo is with the University College of Engineering of Vitoria-Gasteiz, Department of Control Engineering. University of the Basque Country, Spain (email: isidro.calvo@ehu.es)

E. Zulueta, is with the University College of Engineering of Vitoria-Gasteiz, Department of Control Engineering, University of Basque Country, (e-mail: ekaitz.zulueta@ehu.es).

This work has been supported by the University of the Basque Country (UPV/EHU) projects EHU07/40 and PIE09/14

Manuscript received 09 February 2010. Published as resubmitted by the authors 25 April 2010.