A REMOTE LABORATORY FOR AN AUTOMATIC CONTROL ELECTROMECHANICAL SYSTEM

Rafael A. Jordán[1], Alexander Martínez[2], Eugenio Tamura[3], Michael F. Martínez[4] and Carlos A. Olarte[5]

Abstract *— The paper illustrates the design and implementation of a set of electromechanical pieces, electronic boards and software modules that can be assembled in such a way as to build four different ensembles: an electrical elevator, a horizontal displacement system, a servomotor, and a simple pendulum. To interact with the plants there is a software application that enables the user to make adjustments of the parameters of an associated Proportional, Integral and Derivative discrete position or speed controller so that he/she can set the controller constants (kp, ki, kd and the sampling rate) as well as the desired setpoint(s) for the variable to control (position or speed). This could be accomplished from a remote computer via Internet. Through the application the user can see the behavior of the system in a video format and/or display the curves of signals such as position, speed, control effort, and error versus the reference signal.*

Index Terms — Automatic Control, Internet, On-line Laboratory, Remote Laboratory, Web-Enabled Laboratory.

INTRODUCTION

Laboratory sessions play a paramount role in engineering education since they contribute to validate and reinforce the theoretical foundations taught in lectures. Specifically, in the automatic control area the main problem lies in the analysis, design, implementation, and tuning of the controller. Sometimes however, the design, implementation and setup of the mechanical systems to control (the plant) demand considerable effort and money, causing that the time allotted to devise better controllers decreases. Besides that, in the developing countries the lack of resources makes the problem even worse, since the scarcity limits student coverage. To solve this insufficiency of resources, there are several approaches: virtual instruments [1], virtual laboratories [2], remote laboratories [3-6] or combinations between them [7-10].

Running in a computer, a program with graphic panels can emulate the behavior and even the look and feel (the interface) that the real instrument provides, giving birth to the virtual instrument. In the case of virtual laboratories, a model of the plant is translated into a computer program so that the user can interact with in order to estimate the response to a set of stimuli. On the other hand, in remote laboratories the plant exists physically, but, to interact with it, an interconnection mechanism is required. Such a mechanism can be as simple as a serial line or as complex as an Internet link involving many hubs, switches and routers in between.

This paper illustrates the design and implementation of a set of electromechanical pieces, electronic boards and software modules that can be assembled in different combinations to model four different physical systems, which as a whole allow exercising various control strategies. The paper continues with a description of the available plants, the system architecture, an application and finally, results and conclusions.

AVAILABLE PLANTS

The idea is to provide a set of pieces that can be assembled in four different combinations so that with a reduced number of parts it is possible to build models that serve as typical plants, which in turn, are prototypes of basic processes found in the industry, without incurring in the stages of mechanical analysis, design, parts machining and setup.

The chosen plants are: a DC servomechanism to control position and speed, which allows studying the essential mechanism of electromechanical machineries; a horizontal displacement system with multiple stations and the electrical elevator, to apply the basic concepts learned with the first plant; and, a simple pendulum, where the regulation of speed is very critical.

Of all the plants, the servomotor is the simplest one. As shown in Figure 1, it lays on a short base made of wood (piece number one in the Figure) and two aluminum angles (parts number 3 and 4 in the Figure) that are coupled to the motor shaft. The system inertia can be modified attaching steel disks (part number 2 in the Figure) to the shaft and the friction is changed fastening a belt (not shown) attached to the shaft by means of a bushing (marked with the number 5 in the Figure).

[1] Rafael Jordán, Grupo de Automática y Robótica, Pontificia Universidad Javeriana - CALI, rjordan@puj.edu.co
[2] Alexander Martínez, Grupo de Automática y Robótica, Pontificia Universidad Javeriana - CALI, amartin@puj.edu.co
[3] Eugenio Tamura, Grupo de Automática y Robótica, Pontificia Universidad Javeriana - CALI, tek@puj.edu.co
[4] Michael F. Martínez, Grupo de Automática y Robótica, Pontificia Universidad Javeriana - CALI, telemec@puj.edu.co
[5] Carlos A. Olarte, Grupo de Automática y Robótica, Pontificia Universidad Javeriana - CALI, telemec@puj.edu.co

The project was co-sponsored by the Pontificia Universidad Javeriana Cali, Instituto Colmbiano para el Desarrollo de la Ciencia y la Tecnología "Francisco José de Caldas" COLCIENCIAS and the Secretaría Ejecutiva del Convenio Andrés Bello, SECAB. Contrato 008-001/00

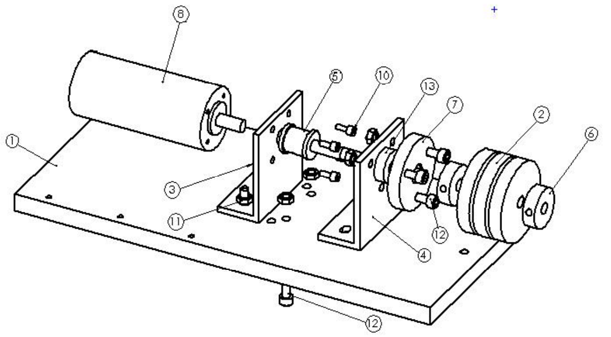


FIGURE. 1
BASIC SERVOMECHANISM FOR POSITION AND SPEED CONTROL.

To build the remaining three plants (horizontal displacement system, electrical elevator and simple pendulum) it is necessary to add other wooden pieces: another base, longer than the previous one, and two triangular angle fasteners. Together, the four pieces of wood form the basic L-shaped framework for any of the other plants. Attached to the longer base, with iron angles, there are two hollow aluminum towers, which are connected by two parallel silver-steel rods. A mobile mechanism, of varying weight, slides over the rods by means of two linear bearings to minimize friction. A servomotor provides the mobile displacement via a transmission belt, which can be tightened to provide better traction. The pieces are joined together using Allen bolts and hexagonal nuts.

The assembly can be used, if resting in its shorter base (in an upright position), as an electrical elevator, or, if stood on its longer base, as an horizontal displacement system; furthermore, in the latter case, it is also possible to add a simple one degree-of-freedom pendulum whose mass can be modified by hanging steel disks. See Figures 2 and 3.

A Minertia FB9M20E motor is used to provide the required power to slide the mobile along the rods. It requires 48 watts and provides 1/15 hp. The motor has an optical encoder coupled to its shaft; it generates 72 pulses per turn, enabling the measurement of position and speed.

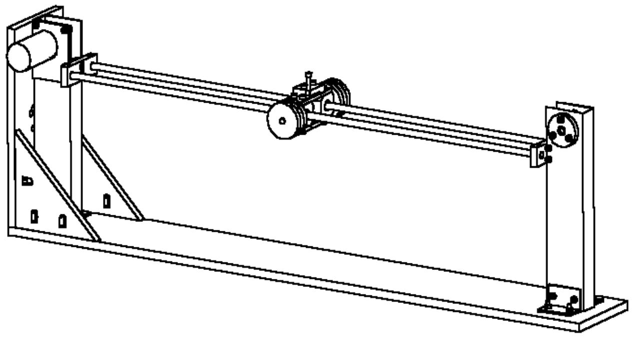


FIGURE. 2
HORIZONTAL DISPLACEMENT SYSTEM AND ELECTRICAL ELEVATOR SETUP.

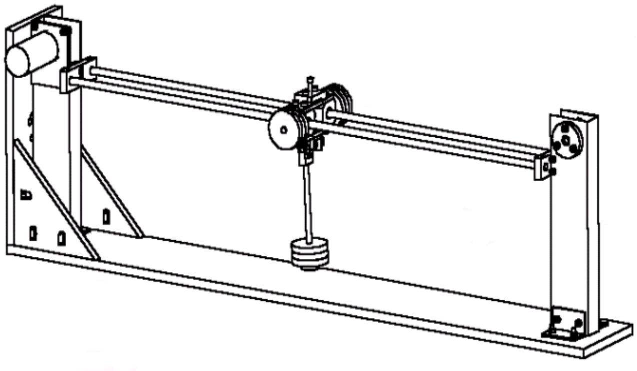


FIGURE. 3
SIMPLE PENDULUM SETUP.

Figure 4 shows the pieces of the set.

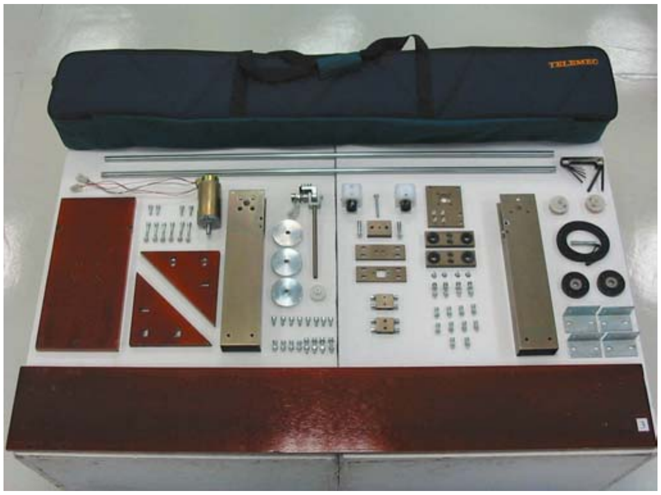


FIGURE. 4
BASIC MECHANICAL COMPONENTS.

System Architecture

To enable the interaction with the electromechanical components the system provides hardware and software modules that allow the interconnection with the processing systems and its teleoperation. Every module was designed as generic as possible so it is feasible to use them as part of other systems where remote interaction is required.

Figure 5 shows a functional diagram of the system: the plant is connected via an interface to a personal computer (PC1), which hosts the controller; the interface provides signal conditioning in both directions. There is also a second processing element (PC2), which hosts the management platform and the teleoperation interface. In synchrony with the acquired measurements, two stand-alone digital video cameras capture the resulting control actions from two viewpoints. These results are displayed in the client computer via a Graphical User Interface, where the user can see the behavior of the system in a video format and/or display the curves of signals such as position, speed, control effort, and error versus the reference signal.

Finally, the two video cameras and the two personal computers are connected to a hub, which in turn is connected to the University network, so that remote users can gain access to the system.

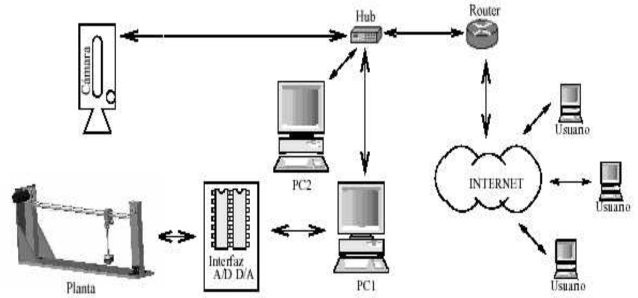


FIGURE. 5
FUNCTIONAL BLOCK DIAGRAM OF THE SYSTEM.

Hardware Architecture

The controller is located in the control computer, a Pentium-based personal computer (PC1); a data acquisition card, connected to the parallel port, provides the I/O interface to the plant so that there is no need to open the computer case. Since the electromechanical equipment demands more electrical power than the data acquisition card can supply, there are two additional boards: one for the sensing elements and the other for the actuators. All of the boards are connected via ribbon cable.

On the other hand, the management and communications server is hosted in other off-the-shelf personal computer, since they offer the best price-to-performance ratio. The two stand-alone video cameras, Axis 2100 [7], provide 10 frames per second and offer two viewpoints, though from a fixed angle.

Figure 6 shows the hardware modules:

- **Power Driver and Sensing Circuitry**: The power driver, shown in the upper center, has differential inputs and outputs and incorporates a 2.2 A over-current sensor that disconnects the load when that limit is exceeded. The sensing circuitry, shown at the lower right, has a frequency-to-voltage converter to measure the rotational speed of the motor, a circuit to determine whether it is a clockwise or counterclockwise spin and a circuit to generate interrupt pulses; the angular position is calculated by counting the pulses from the motor encoder. Besides that, the board is connected to microswitches for out-of-bounds checking, so that the control computer can turn off the power should this happens.
- **Data Acquisition Board, DAQ**: The board, shown in the lower center, has two 8-bit DACs; one 8-bit, 8 single-ended or 4 differential channels ADC. It also provides 16 digital input channels and 16 digital output channels plus synchronization signals.

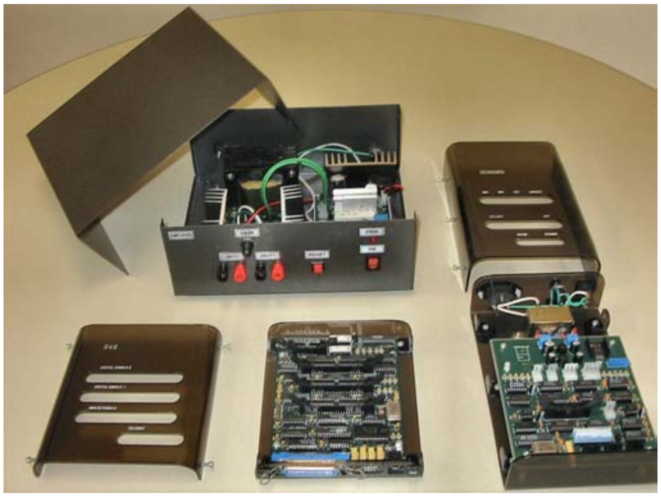


FIGURE. 6
HARDWARE MODULES.

Software Architecture

The software component is composed of three major blocks: the first block has all the basic support routines, the second block provides the routines to allow the teleoperation and the third block has the management routines.

One crucial criterion was to use free software to minimize costs. Therefore, Linux [12] and the GNU tools [13] were selected as the development platform. The GUI for the remote user however, is written in Java to ensure platform independence on the user side.

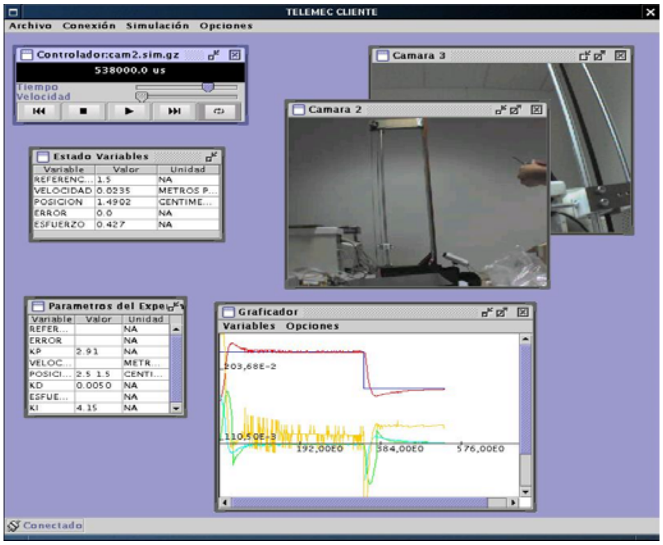
Trough the GUI, the user could modify the parameters of a discrete PID controller (kp, ki, kd and the sampling rate) as well as the setpoints.

- **Basic Support Routines**: Written in C, they provide the code to interact with the data acquisition board. The routines are statically linked to a kernel module, also written in C, which embodies the DAQ interrupt handler and the controller. The reason for using kernel modules is that they have no restrictions and can be dynamically loaded and unloaded to/from memory on demand. Since the DAQ is connected through the parallel port, in order to guarantee a very low interrupt latency time, this module runs on top of RTLinux [14]. However, since RTLinux does not provide library functions to interact with the user, the user application (the front-end controller) needs to run on top of Linux. For this reason, FIFOs are needed to provide the interface between the front-end controller and the controller. Having a front-end controller and the support routines it is possible to command the plant directly from the control computer.
- **System Teleoperation**: The remote operation adheres to the client/server philosophy. So, a communications server, written in Java, validates and stores the client

requirements providing also the interface to the controller. The management facilities are reached via a Web browser over the Internet by using Tomcat [15], a servlet container, written in Java. On the remote user side, the client receives user commands, which, once validated, are sent to the communications server, which in turn, via CORBA [16], sends a requirement to the front-end controller. The front-end controller then communicates with the controller; at the same time, the communications server, by using Java Media Framework, JMF [17], generates an http transaction to the video cameras embedded web servers. Once the communications server has the data from the controller and the images from the cameras, it packs the information so the client can get it, via FTP.

- **Client Application**: To enter into the system the user must open an http session to obtain an account and make a reservation. Afterwards, he/she can use the GUI, a stand-alone Java application implemented using Swing [18]. See Figure 7. The user can:
  - **Enter the system by providing a login and password.** In addition, the user can choose the operating mode: instructor mode, in which it is possible to share experiments with other users; pupil mode, in which he/she just receives the experiment results passively; and single mode, in which he/she is the only current user in the system.
  - **Set up the experiment.** Determine which of the available experiments he/she wants to make and their corresponding parameters. Choose the variables to plot and the cameras to capture the images.
  - **Observe the results.** Plot the data and visualize the video images. Navigate through the data by using a slider bar or the VCR-like buttons. Store the results.
- **Administrative Application**: It is composed by several hhttpServlets provided by Tomcat, allowing that the remote management duties can be done via a Web browser, so that the administrative users can configure and manage the system. See Figure 8. The following are some of the characteristics that can be configured:
  - **System Configuration.** Add plants and experiments. Describe the required parameters for each experiment and its units. Set the upper and lower limits for the parameters. Configure the video cameras.
  - **System Management.** Fill in basic information (like city names, university names) for later user registration. Define operation timetables. Make reservations. Disable/enable the operation of a plant. Disable/enable user accounts.
  - **Operational Configuration.** Configure the FTP server. Set the filename of the controller. Specify the location of the log files. Specify the IP address and port numbers of the video cameras.

The database manager is PostgreSQL [19]; and JDBC [20] provides the connectivity to the database manager. Servlets are used to generate the dynamic web pages.



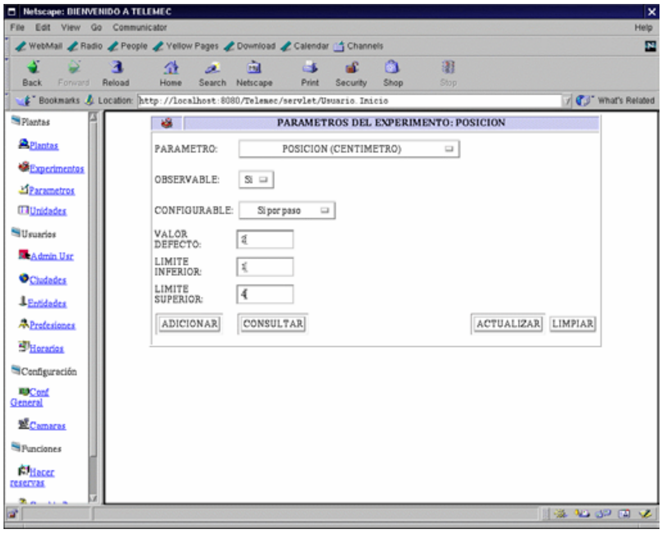
Estado Variables

Variable	Valor	Unidad
REFERENC...	1.5	NA
VELOCIDAD	0.0235	METROS P...
POSICION	1.4902	CENTIME...
ERROR	0.0	NA
ESFUERZO	0.427	NA

Parametros del Expe...

Variable	Valor	Unidad
REFER...		NA
ERROR		NA
KP	2.91	NA
VELOC...		METR...
POSICI...	2.5 1.5	CENTI...
KD	0.0050	NA
ESFUE...		NA
KI	4.15	NA

FIGURE. 7
CLIENT APPLICATION.



http://localhost:8080/Telemec/servlet/Usuario.Inicio

PARAMETROS DEL EXPERIMENTO: POSICION

PARAMETRO: POSICION (CENTIMETRO)

OBSERVABLE: Si

CONFIGURABLE: Si por paso

VALOR DEFECTO: 2

LIMITE INFERIOR: 1

LIMITE SUPERIOR: 4

FIGURE. 8
ADMINISTRATIVE APPLICATION.

APPLICATION

With the system built, during the laboratory practices for the Automatic Control courses it is possible to show some particularly special effects; some worth of mention are:

- Effect of feedback in control systems
- Root locus analysis
- Design of classical controllers
- Evaluation of the impact of disturbances and parametrical changes.

The practices developed included the mathematical modeling of the electrical elevator, shown in Figure 9, and the design and performance evaluation of classical controllers for position and speed.

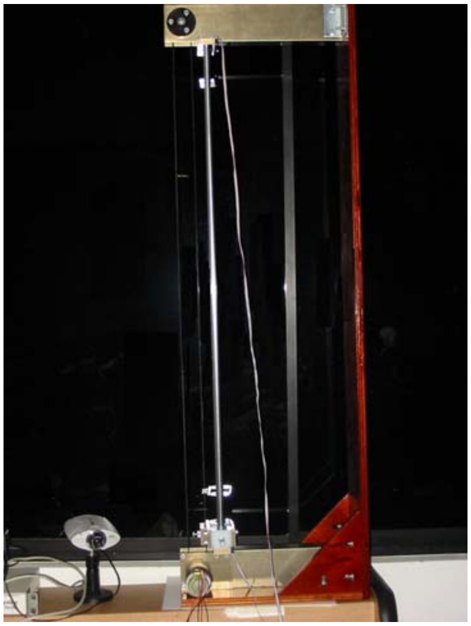


FIGURE. 9
ELECTRICAL ELEVATOR.

To determine the mathematical model of the physical plant the voltage applied to the power driver and the voltage measured on the speed sensor were tabulated, so that the system dynamics was observed and the system identified. The physical study of the mechatronic system and the results obtained during the tests led to linear models for position and speed of the system; these models are:

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{27.92}{0.140s+1} \qquad (1)$$

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{27.92}{0.140s^2+s} \qquad (2)$$

where:
Ea: armature voltage
Ωm: Angular speed
Θm: Angular position

Design of Classical Controllers

To verify the performance of the mechatronic system, PI controllers in continuous and discrete time were implemented, using operational amplifiers in the former case and a kernel module running on top of RTLinux in the latter case.

- **Speed control:** The resulting constants for a PI speed controller were: Kp=5.0, Ki =28.13. The performance of the controller is shown in Figure 10.

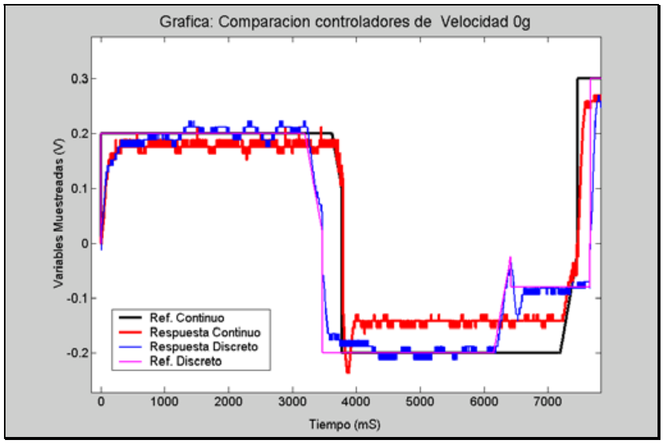


FIGURE. 10
SPEED CONTROLLER PERFORMANCE.

The Figure shows that the performance of the continuous controller is better than that of the discrete one, since there is a lower following error and the characteristics of the transient response are better. It should be noted that during these tests the mobile had not extra weight.

- **Position control:** The resulting constants for a PI position controller were: Kp=2.92, Ki =3.4. The performance of the controller is shown in Figure 11.

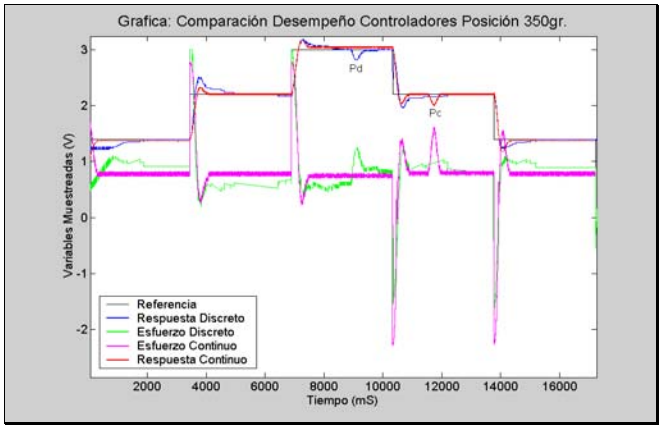


FIGURE. 11
POSITION CONTROLLER PERFORMANCE.

Figure 11 shows a similar behavior between the continuous and discrete controllers with regard to the transient response, the stable state error and rejection to disturbances. It is clear however that the continuous controller demands a little more control effort during the stable state, which indicates that the discrete controller needs less energy to keep the system under control. During these tests the mobile had an additional weight of 350 gr.

CONCLUSIONS

This paper presented the criteria and results obtained during the design of a system, composed of several pieces, in the form of a kit that allows making remote laboratory practices, via Internet, thus increasing the flexibility of timetables and the availability of equipment. As an added benefit, the user can spend more time analyzing the results once he/she has ran the experiment and stored the results in his/her computer.

The open, flexible and versatile design and implementation of the system, is suitable to be used in another areas where remote interaction is required: from the control station the appropriate parameters can be set, and from the system side the report of monitored variables (including video format, if desired) can be generated; having the interface to capture external data, all that is required is that the appropriate standard interface between the application and the controller is used.

ACKNOWLEDGEMENT

The project was finished thanks to the enthusiastic efforts of student and teacher members of Telemec, Grupo de Automática y Robótica. The dedication of the following people merit special mention: María S. Restrepo, Liliana Díaz, Angela M. López, Carlos Vargas, Camilo A. Linares, Jairo A. Osorio.

REFERENCES

[1] Zaimoviæ-Uzunoviæ, N., Lemeš, S. Petkoviæ, D., "Virtual Instruments – A Chance to teach Engineering at a Distance", *International Conference on Engineering Education 2001,* august 2001

[2] Gustavsson, I., "Laboratory Experiments in Distance Learning", *International Conference on Engineering Education 2001,* august 2001

[3] Saad, M., Saliah-Hassane, H., Hassan, H., El-Guetioui, Z., Cheriet, M. "A Synchronous Remote Accessing Control Laboratory on the Internet", *International Conference on Engineering Education 2001,* august 2001

[4] Enloe, C.L., Pakula, W.A., Finney, G.A., Haaland, R.K., "Teleoperation in the Undergraduate Physics Laboratory – Teaching an Old Dog New Tricks", *IEEE Transactions on Education*, 42, No. 3, august 1999, 174-179

[5] Sheng, H., Xu, S., Dalager, B., Kristiansen, V., Strøm, Ø., et al., "Conducting Laboratory Experiments over the Internet", *IEEE Transactions on Education*, 42, No. 3, august 1999, 180-185

[6] Aktan, B., Bohus, C.A., Crowl, L.A., Shor, M.H., "Distance Learninng Applied to Control Engineering Laboratories", *IEEE Transactions on Education*, 39, No. 3, august 1996, 320-326

[7] Gillet, D., Latchman, H.A., Salzmann, Ch., Crisalle, O.D., "Hands-On Laboratory Experiments in Flexible and Distance Learning", *Journal of Engineering Education*, april 2001, 187-191

[8] Naghedolfeizi, M., Henry, J., Arora, S., "Remote Laboratory Operation: Web Technology Successes", *2001 ASEE Southeast Section Conference,* 2001

[9] Saliah, H. H., Villardier, L., Kedowide, C., Assogba, B., Wong, T., "Resource Management Strategies for Remote Virtual Laboratory Experimentation", *30th ASEE/IEEE Frontiers in Education Conference,* October 2000

[10] Overstreet, J., Tzes, A., "Internet-based client/server Virtual Instrument Designs for Real-Time Remote-Access Control Engineering Laboratory", *ACC99-IEEE0280,* september 1998

[11] Axis Communications, AXIS 2100 Network Camera. Axis Communications- Axis 2100. IN URL: http://www.axis.com/products/cam_2100/index.htm

[12] Linux Online Inc., The Linux Home Page at Linux Online. IN URL: http://www.linux.org

[13] Free Software Foundation, Inc., GNU's Not Unix! – the GNU Project and the Free Software Foundation (FSF). IN URL: http://www.gnu.org

[14] Object Management Group Inc.,. Welcome to the OMG's CORBA site. IN URL: http://www.corba.org

[15] Finite State Machines, Inc., FMSLabs – Creators of RTLinux. IN URL: http://www.fsmlabs.com

[16] Apache Software Foundation, The Jakarta Project. The Jakarta Site – Apache Tomcat. IN URL: http://jakarta.apache.org/tomcat/

[17] Sun Microsystems, Inc., JMF Home Page. IN URL: http://java.sun.com/products/java-media/jmf/

[18] Sun Microsystems, Inc., Project Swing (JavaTM Foundation Classes) Software. IN URL: http://java.sun.com/products/jdk/1.2/docs/guide/swing/

[19] PostgreSQL, Inc., PostgreSQL. IN URL: http://www.postgresql.org/

[20] Sun Microsystems, Inc., JDBC Technology. JDBC Data Access API. IN URL: http://java.sun.com/products/jdbc/