REMOTE CONTROL LABORATORY USING MATLAB AND SIMULINK: APPLICATION TO A DC MOTOR MODEL

Rafael Puerto, Luis M. Jiménez, Óscar Reinoso
César Fernández, Ramón Ñeco

Dpto. Ingeniería de Sistemas Industriales, Universidad Miguel Hernández, Elche (Alicante), 03202 Spain

Abstract: In this paper, the general architecture of an application for remote real time execution via Internet of physical process controllers, is presented. This application has been developed using the platform Matlab/Simulink (Mathworks, 2004). The motivation of this work is based on the short availability of real physical systems or laboratories to perform the experiments in control courses. In this way, control lab assignments with various physical processes present in the remote laboratories can be performed. Also, some examples that show the validity and applicability of the presented architecture, are presented. *Copyright ©2004 IFAC*

Keywords: Real – Time Control Systems, Internet Remote Control, Control Education.

1. INTRODUCTION

The work presented in this paper pretends to provide a general architecture for remote execution in real time of physical processes using Matlab platform. The main motivation is, on one hand, the lack of lab scale models or real physical systems, and for other, the little availability of schedules in the laboratories where the students of automatics and process control subjects develop the practical assignments corresponding with the theoretical contents. Therefore, the present application is focused on the real-time remote control of a physical system through Internet and Matlab as teaching and development platform. This application will allow the students to simulate the operation of a controller for a certain physical process using Internet, and the control in real time, of the physical process in question, using the designed controller. In this way, the student is allowed to develop these activities without being in the laboratory where the real system is installed. Also, the application returns to the user all the information relating the performed execution, besides graphic and other data of interest.

Since 1994, engineering educators and others have demonstrated the feasibility of making educational engineering laboratory experiments accessible for student experimentation and learning using Internet (Molly, 2000). References (Molly and Robby, 2000) and (Molly, 2000) shown some guidelines for the features and performance that must have an Internet based process control system. Recently, some authors have performed a lot of advances in the design of remote control laboratories (Díez *et al.*, 2001), (Pastor *et al.*, 2001), (Puerto *et al.*, 2001), (Sebastián *et al.*, 2002), (Bonivento *et al.*, 2002), (Schmid, 2001). In (Schmid, 2001) a real-time control system via Internet is described, and in (Sebastian *et al.*, 2003), a practical environment based on an artificial vision system through Internet has been developed, but the authors don't control the system in real time. For this purpose, the authors use several software tools from different vendors, which can lead to an increase in the development time, and the maintenance cost.

The platform Matlab / Simulink (with some additional toolboxes), has been chosen for the development of this application for several reasons: first, Matlab, Simulink and the necessary toolboxes constitute a reliable, well-known platform and with a lot of technical support, and commonly used in teaching control courses. Second, the time used to obtain the prototype and in the development is quite inferior to the time needed with other tools and platforms (direct programming in a programming language, etc.). Third, this platform provides tools for remote execution of programs, and for real-time execution on a physical system, through a data system acquisition, using a specific control algorithm. The last reason, and not less important, is the great quantity of researchers that use this platform as development tool for simulations and real applications.

2. GENERAL DESCRIPTION OF THE SYSTEM

The general scheme of the application architecture is shown in Figure 1. In this diagram, the hardware and software elements are split in two main blocks: local area where the user works, and remote area where the whole physical system and control elements are located. The detailed elements of local and remote area are the following:

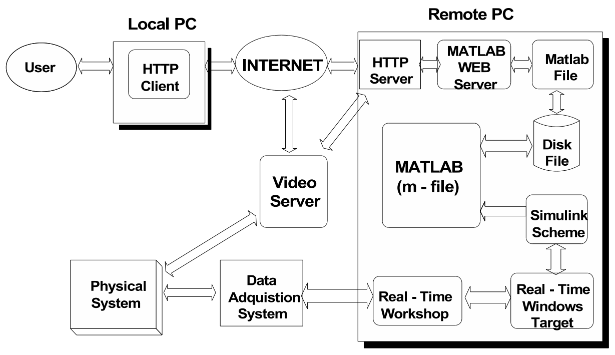


Fig. 1. General Architecture diagram.

- Local area:
  - · Computer with Internet connection and an HTTP 4.0 client application. The Recolab application is optimized for Internet Explorer 6 and Netscape 7 with a minimum resolution of 800x600.
- Remote area:
  - · High speed Internet Connection.
  - · Computer Server: the current implemented system consists on a PC Pentium II (800 MHz) with 256 MB of RAM, giving enough power to run the web server an real – time applications.
  - · Data acquisition system: NI 6024E acquisition board with analog and digital I/O.
  - · Physical system to control: DC motor model 33-002 from Feedback. This laboratory model is made of a DC motor, power amplifier, tachometer, absolute encoder, potentiometer and a magnetic break.
  - · Image capture system and web video server: Sony EVI-D31 / Axis 2400 video server.This video server is based on embedded Linux, and performs video capture for four color video cameras, motion control of cameras, and web video server for static images and continuous video.
  - · Operating system that allows to set access security directives. The system is currently based on Windows 2000 Professional
  - · Http Server Apache v.1.3.27 with PHP 4.3.2 module. This server allows the communication of the computers using the http protocol. Also, it has been configured previously so that the data processing can be made by Matlab Web Server
  - · MATLAB R12 with SIMULINK V. 4.1: executes the program that makes possible the real-time control of the system and the generation of the results in a file.
  - · Matlab Web Server V. 1.2.1: this toolbox allows to use the mathematical and graphic capacities of Matlab and to present the results in a web page.
  - · Real – Time Windows Target Toolbox V. 2.1: this toolbox allows to execute Simulink schemes in real time. For this purpose, it provides the necessary blocks for the interaction with the data acquisition system.
  - · Real – Time Workshop Toolbox V. 4.1: this toolbox generates the C code that, once compiled, will be executed in real time. For space reasons, we don't describe the details of the operation of the Real-Time Windows Target, which is much more complex than the short description presented in this paper.
  - · Control System Toolbox 5.1

2.1 Functionality of Software Application

This application has two aspects clearly differentiated:

(1) Web application: this includes client-server communication using HTTP/HTML protocol, the user interface, users access control, and the main CGI application. The Common

Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. This application is in charge of resource access and communication between Apache and both applications: Matlab and Matlab Web Server.

(2) Real-time execution of control scheme over a specific physical system.

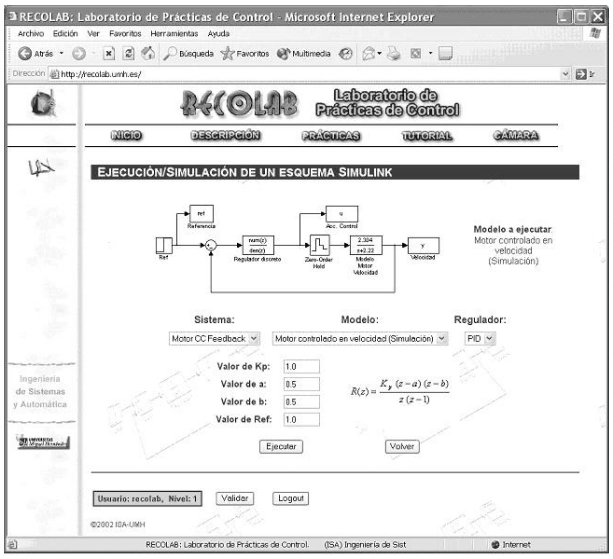
Different programming languages and development tools has been used for each part. Non-critical tasks as user interface, security access and resource sharing, have been coded in PHP (v.4.3.2) running over an Apache HTTP server (Apache, 2004). PHP (PHP, 2004) is a popular script language that has been chosen as far it is an open language widely supported by most web servers and O.S. platforms, and with an extensive library that supports most of network protocols and data base access. PHP code runs on the web sever so it shows a controlled environment for the programmer and can communicate with any other process running in the server (Matlab application in our case).

Of course, PHP as an scripting language is not suitable for real-time applications. The hard real-time core of Recolab (feedback control of physical system) is developed in Simulink with Real-Time Workshop, and compiled with the Real-Time Windows Target tool (Mathworks, 2004).

2.1.1. User Interface of Recolab The user interface of Recolab is based on standard HTML 4.0 markup language. The HTML code is generated dynamically through PHP. The HTML layout is separated from the contents in order to make of Recolab a flexible platform. All formatting of data is based in CSS-styles and PHP layout predefined functions for forms. The data content is stored in configuration text files making quite easy and flexible to add new physical systems, and control schemes without any modification in the PHP code.

The user interface shown in figure 2 is based in HTML forms dynamically generated in PHP from the configuration files. Interactive actualization of forms under user selection, is done through CGI calls to the server and not through client-side scripting. In this way security access can be managed more effectively.

Every control Simulink scheme can be executed in simulation or in real-time over the physical systems using an acquisition board. The user can choose different control strategies and regulators, and can select its parameters accordirng to the simulation experiments been carried out.


RECOLAB: Laboratorio de Prácticas de Control - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Dirección http://recolab.umh.es/
Ir
RECOLAB Laboratorio de Prácticas de Control
INICIO DESCRIPCIÓN PRÁCTICAS TUTORIAL CÁMARA
ref
Referencia
u
Acc. Control
num(z)
den(z)
Regulador discreto
Zero-Order Hold
2.304
s+2.22
Modelo Motor Velocidad
y
Velocidad
Ref
Ingeniería de Sistemas y Automática
Ejecutar
Volver
Validar
Logout
©2002 ISA-UMH
RECOLAB: Laboratorio de Prácticas de Control. (ISA) Ingeniería de Sist
Internet

EJECUCIÓN/SIMULACIÓN DE UN ESQUEMA SIMULINK

Modelo a ejecutar: Motor controlado en velocidad (Simulación)

Sistema: Motor CC Feedback
Modelo: Motor controlado en velocidad (Simulación)
Regulador: PID

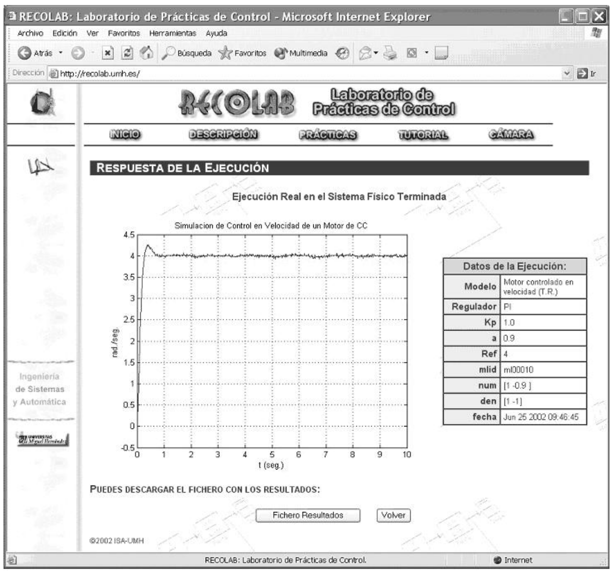
Valor de Kp: 1.0
Valor de a: 0.5
Valor de b: 0.5
Valor de Ref: 1.0

$$R(z) = \frac{K_p\,(z-a)\,(z-b)}{z\,(z-1)}$$

Usuario: recolab, Nivel: 1

Fig. 2. Access control in RECOLAB.

After an execution, the whole sampled or simulated variables of the process can be downloaded. The results page shows part of this information as a graph (the output of the system) (figure 3).


RECOLAB: Laboratorio de Prácticas de Control - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos Multimedia
Dirección http://recolab.umh.es/
Ir
RECOLAB Laboratorio de Prácticas de Control
INICIO DESCRIPCIÓN PRÁCTICAS TUTORIAL CÁMARA
Simulacion de Control en Velocidad de un Motor de CC
rad /seg.
t (seg.)
Ingeniería de Sistemas y Automática
Fichero Resultados
Volver
©2002 ISA-UMH
RECOLAB: Laboratorio de Prácticas de Control.
Internet

RESPUESTA DE LA EJECUCIÓN

Ejecución Real en el Sistema Físico Terminada

Datos de la Ejecución:	
Modelo	Motor controlado en velocidad (T.R.)
Regulador	PI
Kp	1.0
a	0.9
Ref	4
mlid	ml00010
num	[1 -0.9]
den	[1 -1]
fecha	Jun 25 2002 09:46:45

PUEDES DESCARGAR EL FICHERO CON LOS RESULTADOS:

Fig. 3. Results page

2.1.2. Access Control Most of Recolab site can be accessed from Internet without any restriction as far it is a open teaching tool, that could be used by any student interested in control subjet from any University.

The most critical point in Recolab is the Real-Time execution of the Simulink schemes over physical systems. In order to have a more efficient use of the laboratory resources, this processing needs to be done under a validated access.

The validation system used for Recolab is not based in an standard web-server supported system, as far these are focused in control the web resources not the application resources. Instead the validation is coded in PHP so it permits to be integrated with the main application. Rendered options by the PHP code are based on the validation done by user. In this way there is no need to maintain different PHP code for validated or not validated users. If an user request an option that needs validation, the PHP code itself request for user validation.

The validation process is based in a user name and a password. The control access to the resources is based on a user name and a priority level. Most of the resources in Recolab use an access control by level.

2.1.3. CGI Application This application links the input data from the user with the RT control Matlab program, synchronizes the different modules, and records back the output data to the user. To do so, it uses an middle application, Matlab Web Server toolbox.

This toolbox permits, in a flexible way, to adapt the data from the user interface to Matlab. Also, this toolbox can execute matlab functions, and generate graphical web pages with the results. Unfortunately this toolbox is focus on easy deployment of virtual laboratories but not physical remote laboratories. It has not a mechanism to communicate with the Real- Time Workshop and Real-Time Windows Target toolboxes in order to execute real-time control applications.

For the above reason, Matlab Web server is only used for the task of adapt the data from the user interface to Matlab. The rest of non-RT management tasks have been developed in PHP, using files as basic mechanism to communicate the data needed to carry out the real-time control (controller data, etc.), and the results obtained once the experiment has been done. In the next section we comment how this communication is performed.

2.1.4. Real – Time Control Application This application runs the real-time feedback control using the specified control scheme over the physical system. This application has been developed using the following tools: Matlab, Simulink, Real-Time Workshop, and Real-Time Windows Target (Mathworks, 2004). These last two tools allow to generate the real-time code to execute the Simulink schemes using an acquisition board (and therefore, over the physical system, connected to it). Firstly, Real-Time Workshop gives the connection between Simulink and the acquisition board, whereas Real-Time Windows Target allows the real-time excution of the Simulink scheme giving compiled code. The fact of executing a Simulink scheme directly mean an added advantage, as far the complexity and spent time working with Simulink are drastically decreased, allowing an easy and fast design and modification of control schemes.

2.1.5. Real – Time Task and CGI Interprocess Communication As it has been described in a previous section, Matlab software can solve separately the Web based user data communication (through Matlab Web Server), and Real-time execution of Simulink schemes, but it cannot deal with both problems together, i.e. there is not a direct and efficient communication way between Matlab Web Server and the Matlab session running the real – time control task generated by means the toolboxes Real-Time Workshop and Real – Time Windows target.

To solve this problem a PHP-coded CGI starts the Matlab real-time control task with the parameters requested from the user and synchronises its execution using a file-based semaphore mechanism. The whole detailed prodecure implemented to run a Simulink simulation or a real-time execution is as follows:

(1) The user (client) connects to the RECOLAB server and chooses through the user interface the simulation or real-time execution of an specified Simulink scheme. For example, the user can select the type and the parameters of the controller, giving the values of the coefficients of the polynomials that conform the numerator and denominator of its transfer function or the state feedback gain matrix.
(2) PHP module executes the CGI associated to the user interface web page, and sends, to Matlab Web Server, every data related to the execution or simulation to be done.
(3) Matlab Web Server stores the necessary data in a matlab file (.mat) and returns back the control to the PHP module. A file based synchronization mechanism keeps the coherence of data file allowing several concurrent connections.
(4) After Matlab Web Server processing, this application starts the PHP module to run the main control CGI. This CGI is in charge of executing the Matlab control application and resource sharing between the concurrent connections. It also tracks for any problem in the real-time code execution. After checking access privilege, it reads the configuration data of the experiment and waits for the semaphore (waiting that no other session is using the laboratory resources) to start the Matlab session in background. Together with this action the specific M-file is executed under Matlab. This script code reads

the configuration file written by Matlab Web Server and loads the adequate Simulink file, updating its parameters (seleted by the user), and launching the real-time execution or simulation.

(5) Once finished the simulation or the real-time execution, all the result data is stored (included the figures with time response) and the PHP task is signalled, by means a semaphore mechanism, showing that the execution is ended. Main control CGI (PHP) reads the experiment and generates the HTML page with the graphs of the results. In this same page, the user can download the experiment data file (Matlab .mat file type) for later analysis in local computer.

This procedure is shown schematically in figure 4.

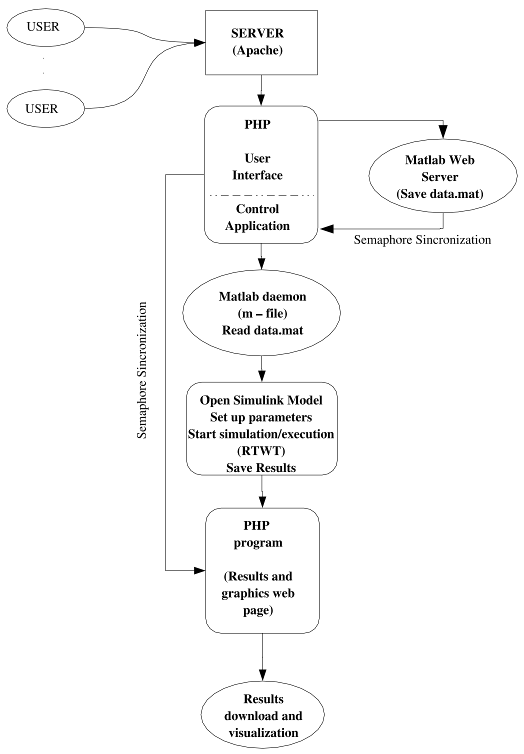

USER
USER
SERVER (Apache)
PHP
User Interface
Control Application
Matlab Web Server (Save data.mat)
Semaphore Sincronization
Matlab daemon (m – file) Read data.mat
Semaphore Sincronization
Open Simulink Model
Set up parameters
Start simulation/execution (RTWT)
Save Results
PHP program
(Results and graphics web page)
Results download and visualization

Fig. 4. Functional description of RECOLAB application.

There must be remarked that only one execution can be done at the same time, as far it must use the physical system and real-time resources. This is why a semaphore control mechanism based in files has been implemented. When more than a request is accepted by the apache web server they are queued waiting until previous excution ends or a timeout expires. Obviusly, only one user can executes a real – time experiment whit the servomotor at the same time. When only a simulation is requested (the system permits also simple simulations of Simulink schemes, indeed its is recommended before a real-time execution) it can be done concurrently.

3. EXAMPLE

In this section some examples of using the system through internet are presented. In the remote area, a physical system composed by a dc servomotor has been chosen. This system has been chosen by their features: easy identification, linear behavior, etc. However the architecture of the system allows to carry out several control experiments with other physical systems with little changes. This is the first advantage of the system architecture.

In this way, the physical system located in the laboratory (remote area)is composed by the following elements:

- The Feedback Mechanical Unit 33-100. The electromechanical components of this unit comprise a dc motor, an analogue tachogenerator, analogue input and output potenciometers, absolute and incremental digital encoders and magnetic brake. In figure 5 this mechanical unit is presented.
- A Data Acquisition Card National Instruments 6024E. The NI–6024E is a high speed analog and digital I/O card for IBM PC/AT and compatible computers. Some interesting features of this board are:
    - · 16 channels (eight differential) of analog input (12 bits resolution)
    - · 2 channels of analog output
    - · 8 lines of digital I/O
    - · Up to 200 KS/s sampling rate
    - · etc.

  Using Matlab and Simulink is perfectly possible to communicate with this board. The sampling time chosen in the system is T=0.01 seconds. This sampling time is sufficient to accomplish the real – time experiments proposed with the dc motor.

So, the purpose is to accomplish some control experiments in the local area with the physical system in the remote area. Nowadays, the control experiments proposed are the following:

- System identification.
- Design of a PID-Controller to command the position of the servomotor.
- Design of a PID-Controller to command the speed of the servomotor.

Each experiment allow to the user in a local area to take advantage of using the physical systems of the laboratories without its physical presence in them. The first task that operator must to carry

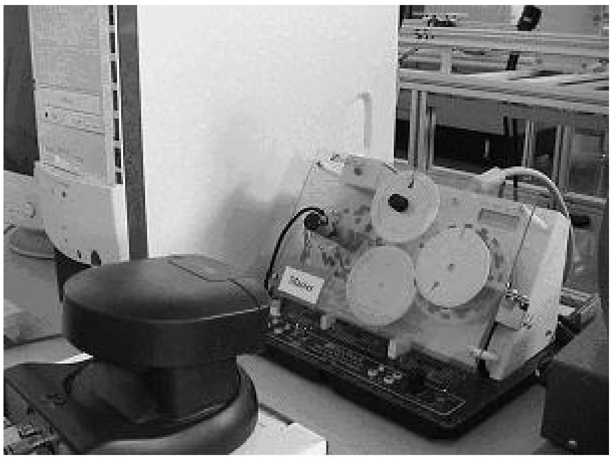


Fig. 5. Feedback Mechanical Unit 33-100

out consist of simulating the designed controllers previous to using them over the real system. So, the user sends to the remote area the data of the designed controllers, and in the remote area through Simulink this system is simulated and analyzed. Once the system is analyzed the user receives de results of the simulation data.

4. CONCLUSIONS

In this paper an architecture that is able to perform real – time executions and to return the results to the user through Internet, has been presented. For this purpose, the Matlab/Simulink platform with the necessary toolboxes was used. The use of this platform is justified, on one hand, for the fact of being broadly used in control engineering, and for the great simplicity, speed and reliability with which it is allowed to develop applications.

Although the example presented in this paper is based on a DC motor model, the general remote control scheme using Matlab can be applied to other physical systems available in the laboratory. The main advantage of the proposed control scheme is that it can be used to do practical work about control theory in remote laboratories in a very simple and quick way. Also, the ideas presented in this paper can make very easy the validation of new control schemes, what allows to the study of different control systems with different physical systems.

The number of completed laboratory sessions in the last academic course (2002/2003) has been considerably increased using RECOLAB system. Specially, it must be note the added flexibility to the laboratory time table. This gives to the student more time dedication to carry out the experimental lessons.

REFERENCES

Apache (2004). The apache software foundation. *http://www.apache.org.*

Bonivento, C., L.Gentili, L.Marconi and L.Rappini (2002). A web based laboratory for control engineering education. *Second International Workshop on Tele-Education in Engineering Using Virtual Laboratories.*

Díez, J.L., M. Vallés, A. Valera and J.L. Navarro (2001). Remote industrial process control with matlab web server. *IBCE 2001. IFAC Proceedings.*

Mathworks (2004). Matlab r12, simulink, real – time workshop, real – time windows target and matlab web server. online manuals.. *http://www.mathworks.com.*

Molly, H. Shor (2000). Remote – access engineering educational laboratoires: Who, what, when, why and how?. *Proceedings of the 2000 American Control Conference. Chicago, IL.*

Molly, H. Shor and Robson Robby (2000). A student – centered feedback control model of the educational process. *30th ASEE/IEEE Frontiers in Education Conference. Kansas City, MO.*

Pastor, R., Sánchez J. and Dormido S. (2001). Related: A framework for publish web laboratory control system. *IBCE 2001. IFAC Proceedings.*

PHP (2004). Hypertext preprocessor. *http://www.php.net.*

Puerto, R., O. Reinoso, R.P. Ñeco, N.M. Garcia and L.M. Jimenez (2001). Remote lab for control applications using matlab. *Internet Based Control Education 2001. A proceedings volume from the IFAC Workshop Madrid.*

Schmid, C. (2001). Virtual control laboratories and remote experimentation in control engineering. *Proc. 11th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering, University of Ulm, Ulm.*

Sebastian, J.M., Garcia D. and Sanchez F.M. (2003). Remote-access education based on image acquisition and processing through the internet. *IEEE TRANSACTIONS ON EDUCATION 46 (1).*

Sebastián, J.M., F.M. Sanchez, D. Santo and R. Aracil (2002). Sivanet: Un nuevo escenario físico remoto para el autoaprendizaje de control a través de internet. *EIWISA'02. III Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática.*