Int. J. Engng Ed. Vol. 22, No. 4, pp. 702–710, 2006
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Virtual and Remote Laboratory for Robot Manipulator Control Study*

ALDO R. SARTORIUS CASTELLANOS, LUIS HERNÁNDEZ SANTANA, ERNESTO RUBIO, IVÁN SANTANA CHING
Universidad Central 'Marta Abreu' de Las Villas, Departamento de Automática y Sistemas Computacionales, Carretera a Camajuani Km 5½ Santa Clara, Villa Clara, Cuba. E-mail: sartorius@uclv.edu.cu luishs@uclv.edu.cu

RAFAEL ARACIL SANTONJA
Universidad Politécnica de Madrid, División de Ingeniería de Sistemas y Automática, José Gutiérrez Abascal 2 E-28006 Madrid, Spain. E-mail: aracil@etsil.upm.es

This paper presents a distance laboratory that enables the remote testing of coupled and non-coupled control algorithms in robot manipulators. The developed laboratory allows the users to use predefined controllers or design their own controllers using the Matlab-Simulink environment. The targeted controllers (designed by the users) can use S-Functions, which they can create themselves using C language; this functionality permits the creation and testing of complex controllers in an easy and fast way. This system uses Matlab-Simulink for practice processing and the Real Time Workshop Toolbox (RTW) such as real time kernel. The laboratory allows simple integration of new robots for control experiments. At present it has a robot manipulator with five degrees of freedom in Cuba, which uses two degrees of freedom for carrying out experiments. The laboratory has been used since year 2003 in identification and control theory in undergraduate courses in Universidad Central 'Marta Abreu' de las Villas and Universidad de Cienfuegos in Cuba, and in robotics and advanced control postgraduate courses in Instituto Tecnológico de Minatitlán in Mexico.

Keywords: distance education; virtual laboratory; remote laboratory; robot control; remote control

INTRODUCTION

ROBOT MANIPULATORS are electromechanical systems described by nonlinear and multivariable dynamic models that offer great challenges for system control design. Further to the theoretical background that new control proposals will have, its impact in the real world will also go through an experimental validation. In the case of a robot manipulator, there are few systems with an open architecture that allow the testing of control algorithms in an easy way; therefore, the control test phase in real robots usually involves a process that takes a lot of time and effort.

The Web-based new technologies have given access to several robotic devices in a remote way through distance laboratories, which can be classified into two major classes: virtual distance laboratories and real distance laboratories. Virtual distance laboratories are based on a physical system simulation in a remote way. Here, through computer animation and the use of specialized software, these robotic devices can be represented in a graphic and an analytical form. Real distance laboratories are laboratories where the users can interact with robotic devices in a remote way.

* Accepted 22 April 2006.

Nevertheless, since robotics is a multidisciplinary field, its study implies the confluence of an array of subjects such as mathematics, mechanics, automatic control, electronics and informatics; thus at present several distance laboratories focused in robotics exist where users can teleoperate robot manipulators or mobile robots through a HTML interface for finding objects [1], visiting museums [2], manipulating objects using a voice system [3] or using vision-based sensing [4-6]; other laboratories allows students to program the task that will execute the robot manipulator in a remote way in many different programming languages [7–12] and some laboratories permit the user to identify the mathematical model of mobile robot [13] or analyze the controller's performance in mobile robots [14, 15] and in robot manipulators [16, 17].

Online robots are a promising aid for education and personal training but at present these systems, when focused in automatic control study, are restricted to using predefined controllers which reduce its ability to choose the gains for any controller (usually a PID controller) in order to evaluate its performance [18]. The complexity in the hardware and software design is drastically increased in distance laboratories that allow students to create and to test control algorithms

in a remote way; these systems must have an open architecture that permit the users to access control structures as well as self-protection mechanisms that prevent the system from being damaged.

In this article the design and construction of a distance laboratory focused on the robot manipulators control study is described. It has been developed in Universidad Central 'Marta Abreu' de las Villas. The Distance Laboratory System (DLS) lets the users modify the trajectories, change control parameters as well as design and test their own control algorithms in a simple way using very well known tools in the automatic control field such as Matlab and Simulink. One of the main features of DLS is that the remote users can create controllers using the blocks given by Simulink, as well as to incorporate S-Functions defined by the users in C language. This laboratory enables testing complex control algorithms using modern techniques of computer-assisted control systems design (CACSD).

DLS ARCHITECTURE

The DLS is divided into three parts: user interface, practices management and practices processing as shown in Fig. 1. The users interact with the system through the Internet. When accessing to the system website, the users must first register by giving their username and password and then choose the practice they want to carry out. There the user can fill all the data in the form associated with the practice in a correct way and finally choose whether to carry out the practice in a simulated or a real way. The CGI located in the webserver receives the data and sends them to the *Practice Management Server* (PMS) as a new order. The PMS verifies which workstation can carry out the practice order and when found, the order is extracted from the list and PMS sends the order to the *Practice Management Client* (PMC) installed in the workstation. When the order arrives at the PMC, the data are processed and the practice is carried out using Matlab-Simulink together with Real Time Workshop (RTW) Toolbox. Once the practice has been processed, the result is transmitted inversely with respect to the order that brought in the practice response. The response is a wep page showing the processing results.

The webserver and the PMS are installed in a computer operating with Windows 2000. For the processing the practices, the PMC and Matlab-Simulink need to be installed in the workstations. The workstations operate within Windows 98.

Common characteristics of distance laboratories

Like other distance laboratories, the DLS has some features in common with most distance laboratories in operation at present.

- *Accessibility*. Since the DLS is mounted on a web platform, the users can access the system from any place in the world. Thus, only one computer connected to the Internet and a browser (Netscape Navigator or Microsoft Internet Explorer) is needed.
- *Ease of use*. For using the DLS the users should only have some knowledge about robotics and control systems, such as kinematics and dynamic robot manipulator modeling as well as controller adjustment and design. In this way the users are focused on reinforcing these topics and avoid all implementation and operation problems of devices used in the practice.
- *Availability*. Web-based learning systems should be available 24 hours a day. This implies that the system should have self-protection rules for accomplishing this requirement. All the experiments must be equipped with hardware and software devices to prevent damage to the components and the people working in the laboratory.

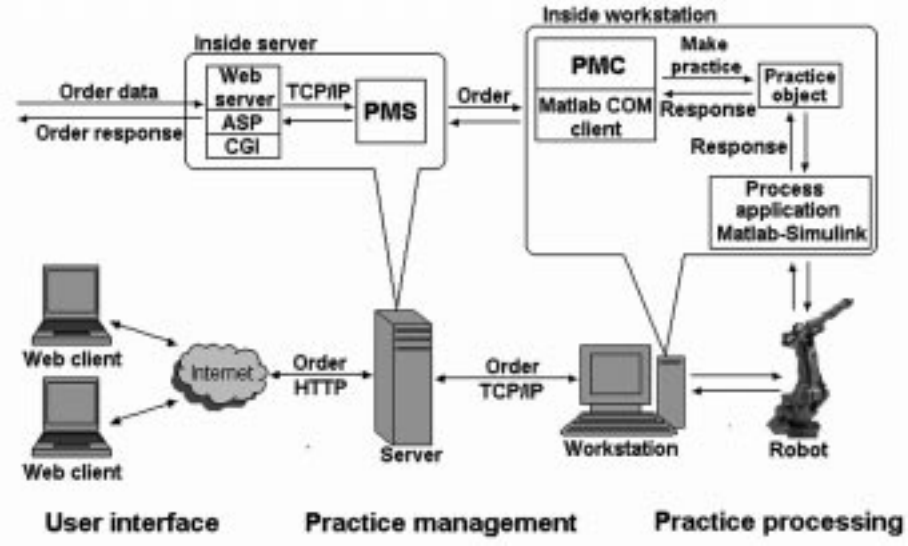

Fig. 1. Software and hardware level architecture of DLS.

Specific characteristics of the DLS

Apart from the characteristics described above, which are common to most of the distance laboratories, the DLS has some additional characteristics which are given below.

- *Easy and fast user interface*. A very important part in the development of a web-based learning system is the user interface. The main function of this part of the system is making the practice order and sending it to the Web server. The DLS user interface is based on HTML pages that use JavaScript and ASP functions; this allows the users to access the system fast and without downloading or installing any additional software. The system also has help pages which provide the user with information about the kinematics and dynamic modeling for the robot manipulators used in the practices, manufacturer data and controller adjustments as well as any other relevant information for the practices.
- *Controller development using Matlab and Simulink in a remote way*: One of the most important features of the DLS is that it allows the users to design their own control algorithms using the Matlab-Simulink environment. These programs are standard tools in the automatic control field, so the users don't need to waste time learning new programming languages for implementing a new control algorithm; they only need some basic knowledge of the Matlab-Simulink environment. Through the Simulink graphic interface, a large number of blocks can be selected and connected easily, allowing the users to create analogical, digital or hybrid controllers very fast. Furthermore, the DLS lets the users, as an option, to create complex controllers including S-Functions defined by the users. These S-Functions must be implemented using the C programming language, a very powerful and widely known language that almost all engineering students are familiar with at present.
- *Controller type*: All the experiments in the DLS can be controlled in two different ways: with predefined controllers or with controllers created by the users. In the first case the users should assign values to the controller gains; for example, the user can choose a PID controller and change the proportional, integral and derivative gains parameters to analyze how they affect the robot manipulator performance. In the second case, the users can design their own controllers (as it has been previously explained) and send it to the DLS where it will be implemented and used for controlling the robot manipulator.
- *Trajectory changes*: The system allows users to change the parameters of the robot manipulator trajectory in order to verify the controller performance in the presence of different input signals.

DISTANCE LABORATORY SYSTEM USE

The DLS user interface has a structure as shown in Fig. 2; in (1) there is the laboratory's user administration zone, in (2) the link buttons to the different parts that make up the laboratory and in (3) the practice visualization zone. When users access the DLS website, they must first register in the system giving their user's name and password in the user administration zone.

Once inside the system, the users can, through link buttons, carry out several operations, such as seeing the theory pages, contacting with the laboratory developers as well as seeing the practices available in them. All these wep pages are located in the system webserver which is common for all the practices. After the user chooses one practice, he is shown a wep page containing the practice diagram block, an explanation about the practice symbols, links to theory wep pages as well as a form where the user can modify the trajectory parameters, the predefined controller parameters or create a new controller defined by the user.

Practices with a predefined controller

In this type of practices the users only need a Web navigator (Netscape, Internet Explorer or

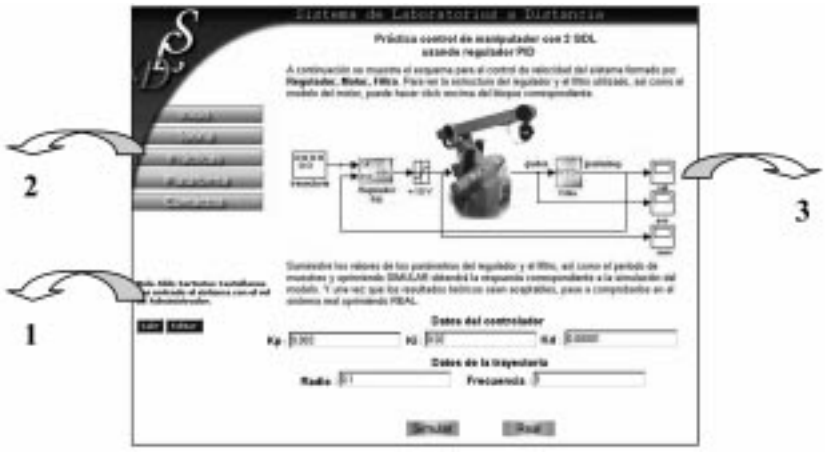

Fig. 2. User interface DLS structure.

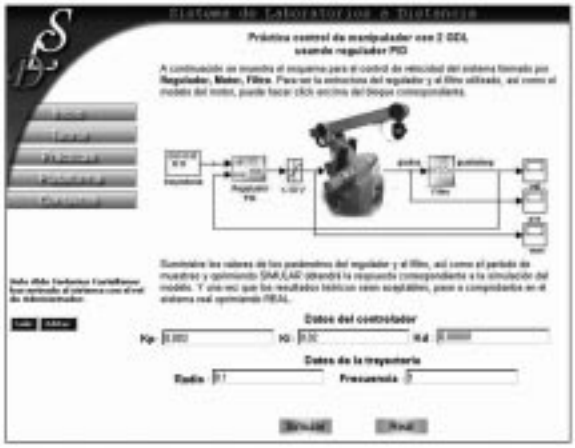

Fig. 3. Wep page and form for carrying out the practice with predefined controller.

any other) to access the DLS website. This type of practice page is given in Fig. 3, which shows the user interface for carrying out practices so as to test decoupled PID controller performance in a robot manipulator with two degrees of freedom. In the form associated to the practice the users can modify the PID controller gains as well as the robot manipulator trajectory data. In addition, there are two options for executing the practice: a simulated one (the execution is simulated and an idealized answer is obtained) or a real one.

Practices with a controller created by the user

When users access practices to create the controller that they will use, they are shown a page such as the one given in Fig. 4.

From this page the user can download a Simulink file containing the practice diagram block. For carrying out these types of practices the users need to have the Matlab-Simulink software installed in order to modify the downloaded Simulink file, as will be explained in the following section.

Fig. 4. Web page and form for carrying out the practice with a controller created by the user.

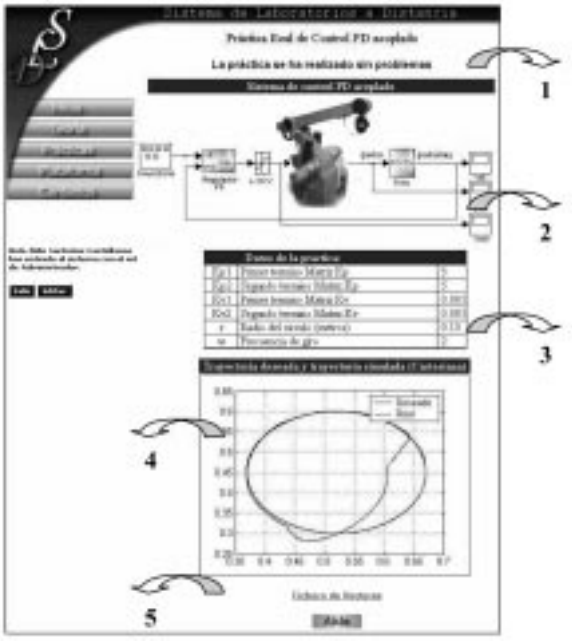

Fig. 5. Practice response web page (summarized).

Response wep pages

The response web page format for all practices is shown in a summarized form in Fig. 5; these pages contain the following fields:

1. *Warnings about practice accomplishment*. In this page section the user is informed if the practice has been successfully carried out; alternatively, he is informed about any error.
2. *System block diagram*. This part of the web page shows the user the block diagram of the robot-controller system used in the practice.
3. *Practice data*. Both the trajectory data used in the practice and the controller parameters are shown here. For practices with controllers created by the users, this section does not appear in the response web page.
4. *Graphics*. The graphics associated with the executed practice are shown in this part. These graphics show the real and desired joint position, the joint position errors and the real and desired Cartesian trajectory that the robot manipulator followed during the practice. Each graphic allows making a zoom for analyzing their data with more detail.
5. *Data file*. In each response web page, there is a link that permits the users to download a text file that contains, grouped by columns, the real and desired joint position, the joint positions errors and the real and desired Cartesian trajectory that the robot manipulator followed during the development of the practice.

CREATION OF CONTROLLERS

One of the most important features of the DLS is the possibility for users to create their own controllers in a remote way. These controllers can be created using only Simulink blocks or, as an option, a combination of Simulink blocks with functions defined by the users. For a better understating, examples are given for controllers created by the user for controlling an Asea IRB-6 robot manipulator.

Controller creation using Simulink blocks

When the user chooses a practice that permits the user to create a new controller, it is visualized by a web page with a form such as the one in Fig. 6. In this page the user must download a Simulink file that contains the control system block diagram (Robot.mdl). In this file the user should modify the controller subsystem using the Matlab-Simulink software, without changing the input and output subsystem names, as shown in Fig. 7, where the user has created a PPI controller inside the controller subsystem.

When the modification is carried out, the user should upload the modified Simulink file to the DLS and decide whether to carry out a simulation or to control the real robot. When the user has requested to control the real robot, the DLS first makes a system simulation and, based on the data obtained from the simulation, several tests are made to determine if the controller can be implemented in the real robot. These tests are focused in three aspects:

1. Verify that the Cartesian trajectory should not excede the robot manipulator workspace.
2. Confirm that the link positions should not exceed the mechanical limits of each link.
3. Analyze, through a signal frequency spectrum analysis, that the system should not exhibit high-frequency oscillations that can mechanically misalign the robot manipulator and/or damage its actuators.

Once these aspects have been determined, the DLS implements the controller, compiles the system using the RTW Toolbox and carries out the practice in real time.

In case the DLS detects that the designed controller represents a risk for the robot manipulator, the user is informed about the cause and is given the simulation graphs and data performed so that he can inspect the system performance.

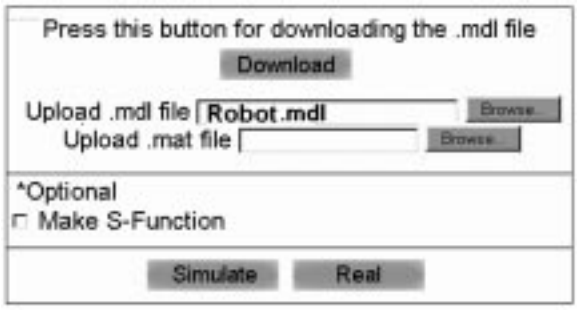

Fig. 6. Form for carrying out practices with controllers created by the user using Simulink blocks.

Controller creation using Simulink blocks and S-functions defined by the user

With this facility the DLS allows the creation of complex controllers which use Simulink blocks and S-functions written in C language. When the user selects a practice with the possibility of creating a controller, in order to make active the S-function creation check box, he is shown a web page that contains a form similar to the one shown

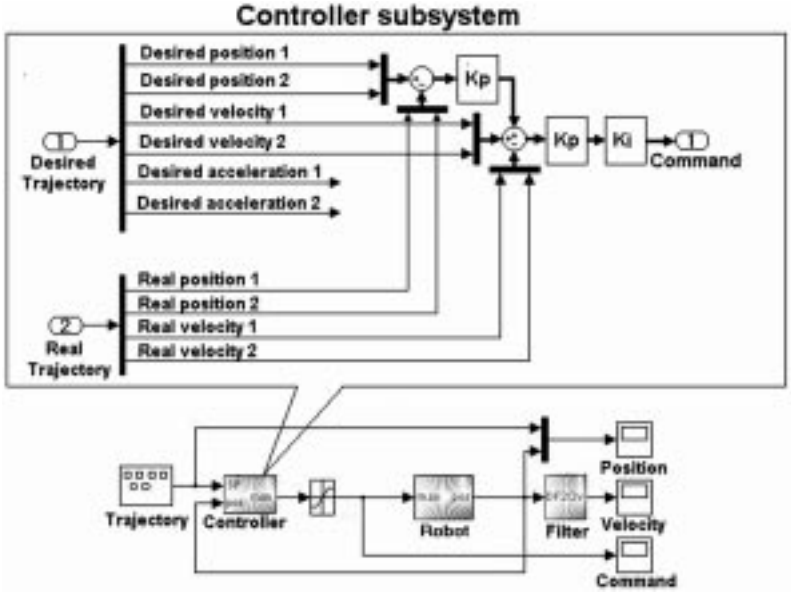

Fig. 7. PPI controller implemented by the user using Simulink blocks for controlling an Asea IRB-6 robot manipulator.

Fig. 8. Form for carrying out practice with a controller created by the user using Simulink blocks and S-functions defined by the user.

in Fig. 6, but with five additional fields as shown in Fig. 8, which are referred to below.

- *Name.* In this field the S-Function name should be specified. This will be the block name defined by the user.
- *Input.* Specifies the number of multiplexed inputs that will enter the block defined by the user.
- *Output.* Specifies the number of demultiplexed outputs that can be realized as defined by the user.
- *Main function.* Here the user should write the S-function code specifying the block inputs defined by the user as u[0], u[1] . . . u[n] and the outputs as y[0], y[1] . . . y[n]. This code should be written in C language.
- *Auxiliary functions.* Here the user can write the auxiliary functions declaration which will be called from the main function. This code should be written in C language.

As in the previous example, in this web page the user should download a Simulink file containing the practice block diagram (Robot.mld). In this file the user should modify the controller subsystem using the Matlab-Simulink software without modifying the inputs and outputs name as shown in Fig. 9, where the user has implemented a PD controller with adaptive compensation proposed in [19]. This controller uses a user-defined function through an S-function called 'Adaptive' in the controller subsystem. In the five additional fields of the form the user must write the S-Function name (Adaptive in this case), the input numbers (10 inputs), the output numbers (2 outputs) and the main function code as well as the auxiliary function code, as shown in Fig. 8.

Once the modification is carried out and the S-Function specified, the user must upload the modified Simulink file to the server and decide whether to carry out a simulation or to control the real robot. When the DLS is commanded to control the real robot, it first makes a system simulation and, based on the data obtained from the simulation, several tests are made to determine whether the controller can be implemented in the real robot. These tests are focused in three ways:

1. Verify that the Cartesian trajectory should not surpass the robot manipulator workspace.
2. Confirm that the link positions should not exceed the mechanical limits of each link.
3. Analyze, through a signal frequency spectrum

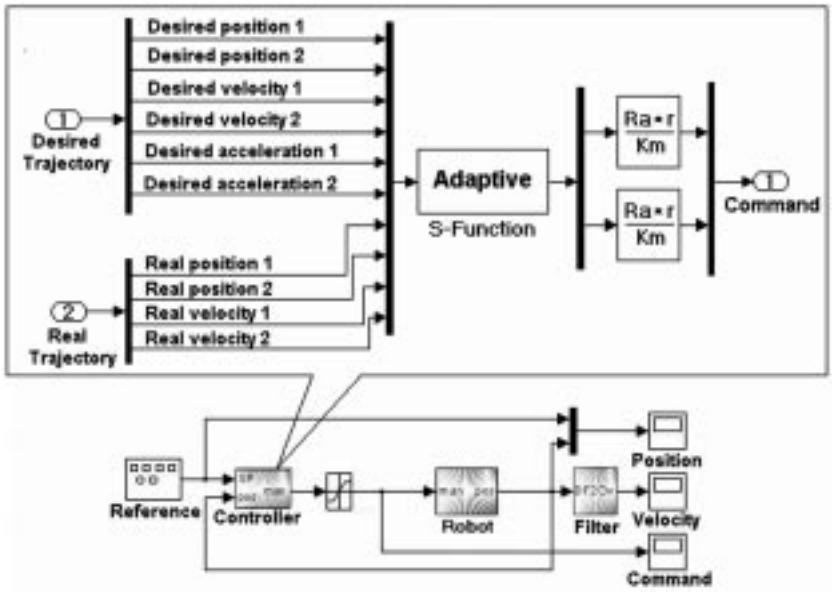

Fig. 9. PD with an adaptive compensation controller implemented by the user using Simulink blocks and S-Function for controlling an Asea IRB-6 robot manipulator.

analysis, that the system should not present high-frequency oscillations that can mechanically misalign the robot manipulator and/or damage its actuators.

Once these aspects have been determined, the DLS implements the controller, compiles the system using the RTW Toolbox and carries out the practice in real time. If the DLS detects that the designed controller represents a risk for the robot manipulator, the user is informed about the cause and is given the simulation graphs and data poduced so that he can inspect the system performance.

TEACHING EXPERIENCE USING DLS

At first, during the first six months of 2003, initial DLS tests from Mexico and Spain were carried out, in order to test controllers in an Asea IRB-6 robot manipulator located in Cuba. While carrying out these test, the system worked satisfactorily but it showed a long delay time between the moment of sending the controller and the moment in which the user received the answer, so we began to work in two directions:

1. Modifying the images format used in the DLS in order to have a faster web page load, as well as to revise the algorithms in charge of implementing the controllers so that the processing time could be reduced.
2. Giving feedback to the user with the robot manipulator data while it is being controlled, using the TCP/IP protocol, so that while waiting for a response web page, the practice state through the control in the Java applet can be viewed. This permits the user to know that the information is being processed and avoids thinking the system is not working.

The modifications referred to in the first aspect have been completed, while those in the second aspect are being developed at present.

Later, the DLS was used in postgraduate courses in the fields of robotics and advanced control theory in Instituto Tecnológico de Minatitlán, Mexico. In these courses, nearly 30 users use the DLS in a simultaneous way to carry out several tests controller practices in the Asea IRB-6 robot manipulator located in Cuba. In the first practice session, a PID controller was used so that the users could evaluate the robot manipulator performance both in the motion control and following trajectory when the different controllers gains vary. After a brief explanation about the theoretical foundations of the practice, the users registered in the system themselves and began to use the DLS in less than five minutes to carry out the practice. In later sessions, the users were able to create and test several control algorithms in a remote way, implemented mainly using the blocks provided by Simulink.

In the present year the DLS has been used in different practices in identification and control courses for Mechanical Engineering and Automatic Engineering students in Universidad Central 'Marta Abreu' de las Villas and in Universidad de Cienfuegos, both located in Cuba. The most stimulating aspects for the students were that they could evaluate the differences that exist between mathematical models and a real robot manipulator and that they could carry out practices not only to any schedule but also an unlimited number of times. These practices have been carried out with small groups (usually 15 students) and, in less than 45 minutes, 15 to 20 practices can be carried out, which shows the high rate in . equipment utilization when using the DLS for remote access.

Using the DLS has allowed the students to understand different aspects of robotics study such as:

1. Analyzing how the controller's design affects the robot manipulator performance both for motion control and for following trajectories.
2. Helping to deepen in the robot manipulator kinematics and dynamic models study and how they can be used, partially or totally, for coupled and non-coupled controller development.

When using the DLS, the users can confirm that they are able to design and evaluate the performance of almost any controller, on an Asea IRB-6 robot manipulator in a remote way, which reduces notably the time and effort required in the experimental validation phase of new controller proposals for this type of device.

At present we are working on the creation of a methodological base that helps us in the construction and use of practices, as well as in the design of a feedback system that allows the users to interact with the DLS in a more effective way, which in turn will allow us to know the impact that the system has in the user learning process. It will also serve as a means for suggesting improvements to the system.

CONCLUSIONS

The Distance Laboratory System (DLS) gives the users the facility of using packets such as Matlab and Simulink together with the RTW Toolbox for the creation and testing of controllers in robot manipulators in a remote way, which allows a reduction in the time and effort in the experimental validation phase for new controller proposals designed for this type of device. Through this functionality it is possible to implement controllers that use complex algorithms which can be easily created using the power and flexibility given by both the C language and the S-functions. Since the DLS software is easily adaptable, the incorporation of new robot manipulators for testing controllers can be fulfilled in a simple way. Owing to the

fact that the system is in a developmental phase, the number of available devices is limited, but we are working towards incorporating new robot manipulators with different configurations to the DLS in collaboration with other universities in the near future. Also, the future incorporation of a better feedback to the user through Java technology will make the system more interactive and more stimulating for the users.

The use of open architecture in the DLS, developed by students from Universidad Central 'Marta Abreu' de las Villas, has remarkably reduced the costs of development and has permitted them to share, with other universities and research centers, a robot manipulator that exhibited a very low utilization rate.

REFERENCES

1. K. Goldberg, S. Gentne, C. Sutter and J. Wiegley, The Mercury Project: a feasible study for Internet robots, *IEEE Robotics and Automation Magazine*, **7**, 2000, pp. 35–40.
2. E. Paulos and J. Canny, Delivering real reality to the World Wide Web via telerobotics, *Int. Conf. Robotics and Automation*, 1996.
3. R. Marin, P. Vila, P. J. Sanz and A. Marzal, Automatic speech recognition to teleoperate a robot via Web, *IEEE Int. Conf. Intelligent Robots and Systems*, 2003,Vol 2, pp. 1278–1283.
4. S. You, T. Wang, R. Eagleson, C. Meng and Q. Zhang, A low-cost Internet-based telerobotics system for access to remote laboratories, *J. Advanced Engineering Informatics*, **15**, 2001, pp. 265–274.
5. H. Hu, L. Yu, P. W. Tsui and Q. Zhou, Internet-based robotic system for teleoperation, *Int. J. Assembly Automation, special issue on Internet & online robots for telemanipulation*, **21**, 2001, pp. 1–10.
6. G. T. McKee: An Online robot system for projects in robot intelligence, *Int. J. Eng. Educ.,* **19**(3), 2003, pp. 356–362.
7. F. A. Candelas, S. T. Puente, F. Torres, F. G. Ortiz, P. Gil and J. Pomares, A virtual laboratory for teaching robotics, *Int. J. Eng. Educ.,* **19**(3), 2003), pp. 363–370.
8. A. Bicchi, A. Caiti, L. Pallottino and G. Tonietti, Online robotics experiments for tele-education at the University of Pisa, *Int. J. Robotic Systems,* 2004.
9 S. Bottazzi, S. Caselli, M. Reggiani and M. Amoretti, A software framework based on real time CORBA for telerobotics systems, *Int. Conf. Intelligent Robots and Systems*, 2002.
10. F. Torres, S. T. Puente, F. A. Candelas and J. Pomares, Virtual laboratory for robotic and automation, *Proc. IFAC Internet Based Control Education*, 2001.
11. D. Austin, Design a Web-based tele-programming interface for mobile robots, *Australasian Conf. Robotics and Automation,* 2002, pp. 206–211.
12. J. L. P. Molina, A. G. González and J. L. Coronado, Modulo de teleoperación para acceso vía Internet a dispositivos robóticos en laboratorios remotos, *EIWISA 2002.*
13. C. Röhrig and A. Jochheim, Java-based framework for remote access to laboratory experiments, *IFAC/IEEE Symp. Advances in Control Education*, 2000.
14. A. Jochheim and C. Röhrig: The Virtual lab for teleoperated control of real experiments, *IEEE Conf. Decision and Control*, 1999.
15. C. Messom and R. Craig, Web based laboratory for controlling real robot systems, *Proc. Biannual Conf. Distance Education Association of New Zealand*, 2002.
16. H. H. Hahn and M. W. Spong: Remote laboratories for control education, *Proc. 39th IEEE Conference on Decision and Control*, 2000, pp. 895–900.
17. C. Hopp, S. Stoll and U. Konigorski, Remote control design and implementation using the Internet, *World Automation Congress*, 2002.
18. A. Casals, J. Aranda, J. Fernández and M. Frigola, Evaluation of learning technologies in robotics, *Int. Conf. Robotics and Automation*, 2002.
19. J. J. Slotine, ed., *Applied Nonlinear Control,* Prentice-Hall (1991).

Aldo R. Sartorius Castellanos graduated from Universidad Veracruzana, Mexico (Electromechanical Engineering) in 2000. He has an M.Sc. in Automatics (Computational Systems) from Universidad Central 'Marta Abreu' de Las Villas. He is currently completing a Ph.D. in Automatic Control at the same University. At present he is the head of the Computer Integrated Manufacturing Laboratory at Instituto Tecnológico de Minatitlán. His interests include adaptive control systems, remote laboratories and robotics.

Luis Hernandez Santana is the leader of the Mechatronics Research Group at Universidad Central 'Marta Abreu' de Las Villas. He is in charge of the University Collaboration Program between Universidad Central 'Marta Abreu' de Las Villas and the Council of Flemish Universities from Belgium. He graduated from Universidad Central 'Marta Abreu' de Las Villas (Automatic Control Engineering) in 1981 and holds a Ph.D. in Science and Technology from the same University.

Rafael Aracil Santonja is Full Professor in Departamento de Automática, Ingeniería Electrónica e Informática Industrial at Universidad Politécnica de Madrid. He graduated (Electrical Engineering) from this University in 1971 and received a Ph.D. in engineering from the same University in 1975. He is currently working in the development of robots for new applications and for intelligent teleoperation. He is the leader of the IFAC Computer Vision Spanish Group and of the Latin American Robotics Network. He has worked for several EU-funded projects (ESPRIT, BRITE, and EUREKA). His interests include automation, robotics and image processing. Dr Aracil has published several books and articles on the above mentioned topics.

Ernesto Rubio Rodriguez graduated (Automatic Engineering) from Universidad Central 'Marta Abreu' de Las Villas in 1997. He received an M.Sc. in Automatics (Robotics and Intelligent Control) from the same University in 2000. At present he is working on his Ph.D. in robotics (pneumatic articulation) at Universidad Central 'Marta Abreu' de Las Villas in collaboration with Universidad Politécnica de Madrid.

Ivan Santana Ching graduated (Automatic Engineering) from Universidad Central 'Marta Abreu' de Las Villas in 1999. He received an M.Sc. in Automatics (Computational Systems) from that University in 2004. His main interests are distance laboratories, data basis and Internet programming.