

Experiences with Virtual Environment and Remote Laboratory for Teaching and Learning Robotics at the University of Alicante*

FERNANDO TORRES, FRANCISCO A. CANDELAS, SANTIAGO T. PUENTE, JORGE POMARES, PABLO GIL, and FRANCISCO G. ORTIZ

Automatics, Robotics and Computer Vision Group, University of Alicante 03080, Alicante, Spain

E-mail: Fernando.Torres@ua.es

This paper begins with a description on the state of the art of robotics education in Spain. We then present our virtual environment and remote laboratory, whose principal component is RoboLab, a system that allows the student to interact with simulated and real robots through the Internet. Afterwards, we describe the main aspects of our Robotics courses at the University of Alicante, as well as the practical exercises carried out by students using the virtual laboratory. We then present an analysis of our experience of using the virtual and remote laboratory and give our conclusions, in which we analyze the advantages and disadvantages of using our virtual environment and remote laboratory to teach Robotics.

Keywords: virtual laboratory; remote laboratory; e-learning; robotics

INTRODUCTION

THE TEACHING of practical courses in Robotics requires traditionally expensive equipment such as robots and their controllers, and often this equipment is inadequate for simultaneous use by many students [1]. Moreover, this equipment is vulnerable to damage if improperly used. Another problem with having to use a laboratory with the appropriate equipment to carry out the practicals is that students need to use these laboratories according to some strict schedules [2, 3].

Nowadays, new technologies such as the Internet, tele-operation of systems and virtual reality, allow students to carry out practical exercises remotely (e.g. from home), so avoiding this problem, by using virtual laboratories [4–8]. Other advantages of these remote systems are that they can offer group-working and self-evaluation mechanisms [9, 10]. However, it is also important to evaluate aspects such as the acceptability of such laboratories by the students, the effect they have on teaching, the accessibility to the new technologies in the local social environment, and the disadvantages as compared with traditional teaching methods [11, 12].

ROBOTICS EDUCATION IN SPAIN

At present, Spanish universities can establish their own academic curriculums and offer two

different types of degrees: an Official Degree, which is valid throughout the country, or a degree that is specific to the university.

Official Degrees are approved by the Government following the proposal of the Council of Universities. The Government also establishes the general guidelines for these degrees in the same way. These guidelines define the minimum homogeneous content, or ‘Basic Subjects’, that the plan of studies for each degree must have. Each university can decide and approve its own plan of studies according to its criteria and the necessities of its social environment. In addition to the ‘Basic Subjects’, these plans of studies also include ‘Compulsory’ subjects, ‘Optional’ subjects and ‘Extra-curricula Activities’. The ‘Compulsory subjects’ are proposed by each university with the aim of specializing its study plans. On the other hand, students can choose different optional courses, corresponding to ‘Optional Subjects’ and ‘Extra-curricula Activities’.

Furthermore, the Spanish university educational system can be divided into two different phases: the ‘long’ or ‘advance cycle’, and the ‘short’ or ‘technical cycle’. In technological studies, degrees are administered in centers called Technical or University Schools. Students study for degrees that contain subjects related to Robotics at these centers.

More specifically, the only official Spanish University degree that has any basic subjects related to Robotics is the ‘Automatics and Industrial Electronics Engineering Degree.’ The only course that includes any basic Robotics subject is

* Accepted 28 February 2006.

'Control and Programming of Robots,' whose basic contents, defined by the Government, are the following: Modeling, Control and Programming of Robots, Task Planning, and Interaction with the Environment. This basic subject must be developed and taught within a minimum time of 60 hours. There are other degrees in the Spanish University, however, for which the student may study subjects related to Robotics, although this is entirely optional. Thus, in the majority of Spanish universities in which Industrial or Computers Science Engineering degrees are offered, there is at least one optional course in which the general aspects of Robotics are studied, such as Control, Programming, Mechanics, Sensors or Navigation.

THE VIRTUAL LABORATORY

The virtual and remote laboratory used for the teaching of Robotics comprises two tools, which have been developed completely by the authors of the paper thanks to the financial support of the national and regional governments and economic aids of the University of Alicante [12, 13]. These tools are called RoboLab and Visual, and can be accessed from <http://disclab.ua.es/roboLab/labvir.htm>.

RoboLab

RoboLab is the main tool of our virtual laboratory. It is a system that allows the students to work with a simulation of an industrial robot and carry out operations with a real robot through tele-

operation; it is described in more detail in [3]. The only facilities that the student requires are a computer that is connected to the Internet, a Web client program and the necessary software components, as the Java runtime libraries [14]. With these the student can access a Webpage with a Java applet that affords access to the functions offered by RoboLab.

Figure 1 shows set up of the whole system. With regard to the laboratory equipment, the only hardware that require any considerable investment are the robots together with their controllers. At present, we are using two robots, a Scorbot ER-IX (Intelitek) with 5 DOF and a gripper, and a PA-10 (Mitsubishi) with 7 DOF plus a tool, although the system is able to manage other robots. Both robots can be simulated and tele-operated, but the first one is usually used for the practical exercises.

The remaining pieces are commercial items that are readily available on the market. The 'main server' is a PC that provides the Web services from the Internet, and supplies the Java applet with the user interface as well as the self-evaluation questionnaires. It also manages the user authentication and his account. A 'tele-operation server' is basically a program that runs on another PC within the private network. It validates the commands to the robot received from a user's computer, translates them to the appropriate language and sends them to the robot's controller. It also obtains information about its current state to allow an on-line feedback. In addition, there is a 'video server' that gives the option of a video stream feed-back for remote operation.

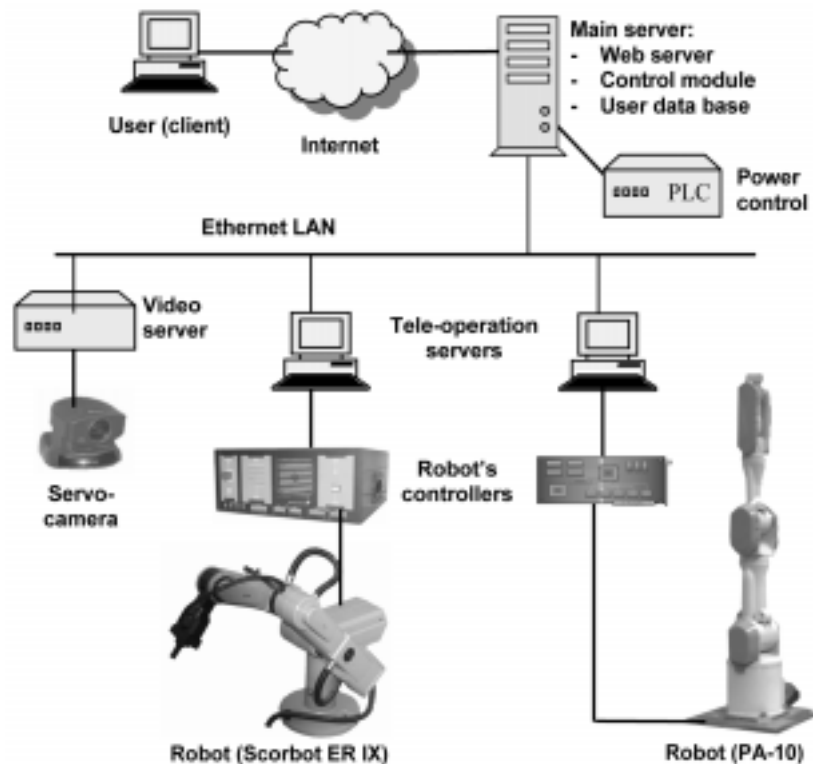


Fig. 1. Hardware architecture of RoboLab 2.

Using RoboLab

The user interface is a Java client applet that can be downloaded by the student from the 'main server' and provides a virtual reality simulation with the three-dimensional representation of the robot-setting and its workspace (Fig. 2, left). In the first version of RoboLab, the simulation is based on VRML (Virtual Reality Modeling Language) [15]. To facilitate the software installation, avoid software incompatibility problems and offer the students a simpler interface, another version (RoboLab 2), which carries out the simulation thanks to Java 3D library from Sun has been developed (Fig. 2, right) [16]. This latest version has many other interesting features, such as those described later in the next section.

The client applet displays a more realistic simulation than the majority of the proposed Java-based systems for simulating robots [17, 18], and the simulated robots can be very similar to the real robots. It also offers a user interface that is very user-friendly for students who are beginners in Robotics.

The range of use of Java-based tele-operation applications is extensive thanks to the portability of this language. Many of the applications for Robotics are for simulating or tele-operating educational equipment [9, 10, 19] or simple robots [17, 20, 21, 25]. In the field of industrial robot arms there are fewer applications [8, 22, 23], and, in general, they are designed for specific robots. Moreover, very few of them are based on an open architecture, which offers the required flexibility to change the robot that is being used or to add new robots without modifying either the user-interface or the architecture of the system [7, 24]. With regard to the simulation, there are not many Java-based applications for industrial robots that offer a realistic virtual environment, and the majority represents only wired-models or simplified structures.

The main feature of our system, in contrast to the others, is that it offers the flexibility of mana-

ging different robots or including new robot models and equipment, as well as other kinds of passive objects in the workspace, without changes in the user-interface and the system's architecture. The developed system also has other interesting features, such as a feedback to the user based on updating a graphic simulation while the robot is moving, the use of high-level communication protocols, and the modeling and virtual representation of the robots, passive objects and scenarios based on Java 3D. This system displays a more realistic simulation than the majority of the proposed Java-based systems for simulating robots, and the simulated robot can be very similar to a real robot. It also offers a user interface that is very user-friendly for operators who are not specialists in Robotics. This is an advantage of our system over that of Matlab robot modeling tools, which require more knowledge of the system to perform the simulation.

First the student performs the exercises on the simulated virtual environment and then, after checking that the results are correct, they can execute them in the real system by means of the tele-operation option. Thanks to the simulation, the students are able to practice and carry out correct movement sequences. Once a correct simulation has been effected, the student can request the 'main server' to remotely execute the movement sequences with the real robot. Although the user interface is always available for executing simulations in the virtual environment, the student must identify himself as a user authorized to use the tele-operation capabilities. The 'main server' also manages the access of several users to the robots, guaranteeing their orderly access.

To verify the movement of the real robot, the client applet has different feedback options, two of which are outstanding. The first one is a compressed video stream, generated by the 'video server' and sent to the student's computer while the robot is executing the movement sequence. This is the usual method of feeding back to the human

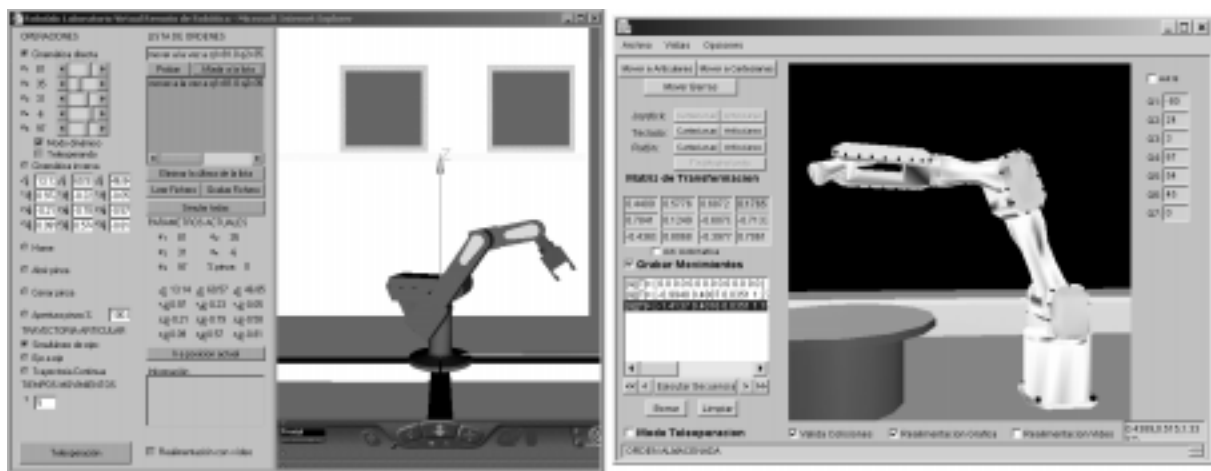


Fig. 2. The two versions of the applet with the RoboLab user interface.

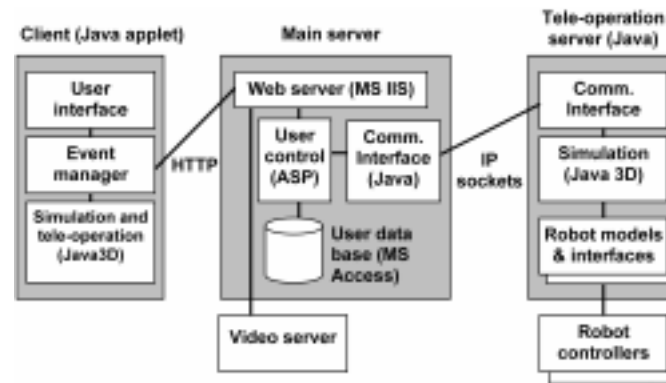


Fig. 3. Functional structure of RoboLab 2.

operator used in remote operation systems [25, 26]. The second is that the simulation in the student's computer is updated with information about the current state of the real robot's joints received online from a 'tele-operation server' [27]. This novel option allows a fluid tele-operation with a lower bandwidth requirement than the first one.

Software architecture of RoboLab

The main blocks of the system's software modules are shown in Fig. 3. There are three main blocks: the client applet with the user interface; the main server, and the 'tele-operation server'. In reality, it can be more than one client applet, and one or more tele-operation server can be running on one or more PC in the laboratory.

The most important feature of the latest version of RoboLab, in contrast to others systems, is the flexibility for users to define new robot models and include them in the client-applet and the tele-operation server. In this way, new robots can be incorporated into the laboratory equipment to be tele-operated, without changes in the user-interface and the system's architecture. To model a robot, we have created, from the Java 3D API, a library with a set of different Java classes [16, 28]. These classes allow us to define the structure, the graphical representation and the kinematics model of the desired robot, as well as other parameters [24]. At present, this library does not take into account the robot's dynamic model. Apart from the robots, it is also possible to define models for scenarios and passive objects in the robot's workspace.

The simulation engine, which is based on our library of Java-3D classes for modeling robots, not only is used in the client applet, but also in the tele-operation servers (see Fig. 3). In this case, the scenarios and objects represent the real robot's workspace so that any possible collisions can be correctly detected. This guarantees a good use of the real system and increases its useful life-span, because only if the simulation is correctly done, are the commands translated to the robot language and sent to the robot's controller.

Other important improvement is that the communication between a client-applet and the main server is based on the protocol HTTP of TCP/IP

architecture. Although this is a very high-level protocol, it allows the communication independently of the networks and firewalls to be crossed.

Improving the user interface of RoboLab with a joystick

Among the many devices that can be used as a user interface for controlling the remote robot arm in Tele-robotics, the force-feedback joystick is the most commonly employed, not only because it is a very intuitive way of specifying the movements that are to be carried out, but also because it transfers the robot arm's sense of touch to the human operator's arm. However, force-feedback joysticks that are made specifically for Robotics are very sophisticated and costly devices.

Taking this into consideration, we have improved the applet for the RoboLab user-interface by giving it the capability of using a commercial joystick designed for computer games, in addition to the typical keyboard or mouse devices. Commercial joysticks for computer games are inexpensive and many models even have simple features for force feedback. However, it should be remembered that many commercial joysticks for games are designed for specific operating systems and their programming requires specific APIs (Applications Programming Interfaces). In contrast, our RoboLab features the portability of applications created with Java. Thus, it is not difficult to include generic joystick capabilities in a Java application, as will be shown here.

The software architecture that we have employed in order to be able to use a joystick with RoboLab is shown in Fig. 4. This architecture, and the software libraries developed, can also be used with other Java applications apart from RoboLab. We have chosen the Microsoft Sidewinder joystick because it is an economical extended model. To access the joystick's functions, we use the M.S. DirectX libraries. DirectX makes the use of the joystick, with any game device, transparent to higher software levels in an M.S. Windows platform, facilitating the programming and the compatibility of the applications for many different game devices.

The problem arises when a Java application has to use DirectX functions, since Java cannot access

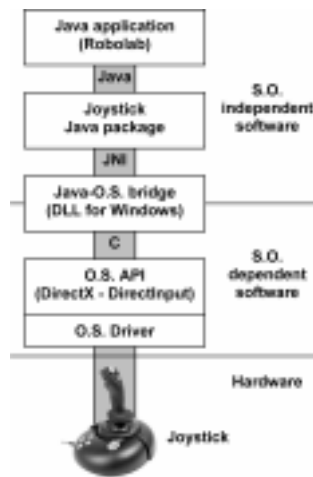


Fig. 4. Software architecture for using a joystick with RoboLab.

this library directly. To solve this problem, a 'bridge' is required between DirectX and the Java package that manages the interaction between the joystick and the application. We have developed a bridge based on a DLL (Dynamic Link Library) for Windows, which is written in the C language.

The Java package that controls the joystick and interacts with the application in a transparent and O.S.-independent way runs over the DLL. This package has to be included as a thread in the application, (i.e., the client applet of RoboLab in our case). The communication between this package and the C-implemented DLL is made through the JNI (Java Native Interface) specification [14].

The Java joystick package has three basic functions. First, it accesses the joystick, to get basic information (i.e., the number of axes and buttons, the model, the force-feedback capabilities, etc.), and to initiate the joystick. Second, it reads the state of the axis and the buttons of the joystick, pre-processes the data obtained and stores it in local structures. Finally, it informs the application's modules when the state of the joystick changes by means of events.

With the software architecture described above, a commercial games joystick can be used with RoboLab to move the simulated robot easily and directly. Moreover, if a force-feedback joystick is used, RoboLab can transmit the contacts or collisions of the robot-arm with its environment to the user, through a sensation of resistance, through the joystick.

RoboLab can use two different sources of information to identify a collision. One is the simulation engine in the client applet, which detects the collision of the robot-arm with either objects that are within the virtual environment or with itself. Furthermore, a force sensor can be incorporated into the robot's end tool to obtain the real haptic information, while manipulating objects, and transmit it to the user interface through the Internet. The client applet evaluates and combines the data from the two sources and controls the

joystick's movements. For the second case, a new software module is required in the 'main server' of the RoboLab architecture (see Fig. 1) in order to obtain, process and send the information from the robot's force sensor through the Internet.

The second method only works when RoboLab is used to tele-operate the robot on-line, in contrast to the command line-option. It should be pointed out that this method only works well when the network delays between the client applet and the 'Web server' are minimal. This can be achieved when the user's computer is connected to the same LAN (Local Area Network) as the 'main server'. In other cases, however, the network delays can make the tele-operating unstable, due to the delay between the user's action and its perception [29].

Projects under development

As already described in the previous sections, our new version of RoboLab can incorporate scenarios and passive objects in the simulation in run-time. It also offers simulation and tele-operating for several robots rather than for one in particular. Currently, in simulation or tele-operation from a client applet, only one robot can be operated at a time, and objects to be incorporated in the simulation are virtual models previously created and stored in the 'main server'. We therefore continue developing the system, to exploit the capabilities of RoboLab better and to overcome the cited limitations, as we now describe.

On the one hand, we are working on the updating of the positioning of passive objects by the client applet, with real information from the workspace in the laboratory, so that the user can command the robot to manipulate and interact with objects. To achieve this, the first stage is the three-dimensional recognition of basic objects in the workspace by relating them to basic three-dimensional models stored in a database (for example, a polyhedron or sphere). The information about the recognized object will then be sent to the simulation engine in the client applet (and to the simulation engines in the 'robot servers'). Finally, the client applet will update the graphic representation. This important feature has already been considered by some authors [17].

Furthermore, we are enhancing the system to allow the user to control more than one robot with the simulation in the client applet, so that the virtual environment is updated with the state of all robots located within the working area of the laboratory and not just the tele-operated one. Along these same lines, we would also like to enable the user to manage more than one robot through the simulation. With such enhancements, the system will then be able to simulate and operate a co-operative robots environment, which will be a novel feature in these sorts of systems.

Apart from the two above-mentioned objectives, we are also improving the user interface of RoboLab with new features designed specifically for handicapped users. The first one is the fusion of

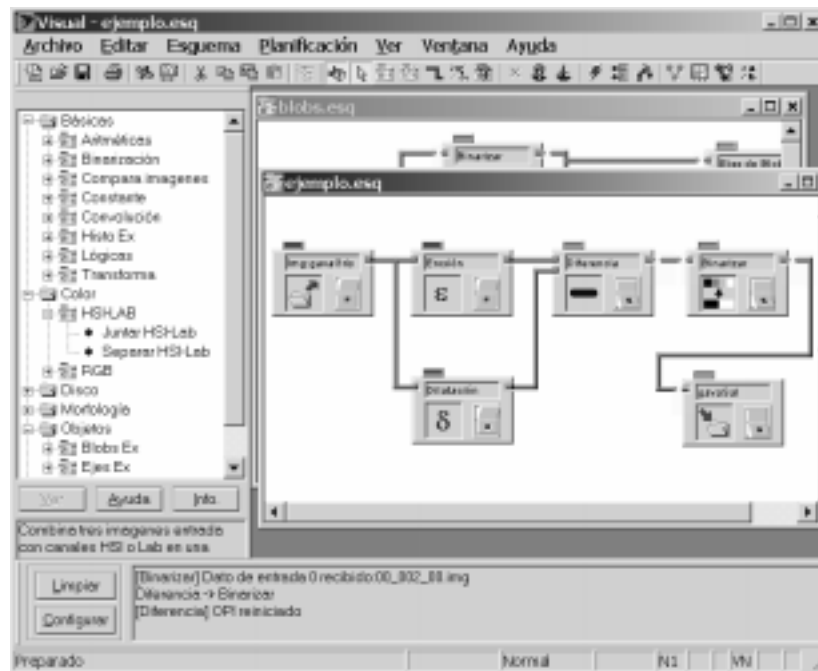


Fig. 5. Visual user interface.

the information from the simulation engine and the video feedback in the client applet's user interface, so that the more relevant data, such as manipulated objects or robot axis positions and orientations, are highlighted in the video window. The second feature will be the inclusion of other human-machine interface techniques in the client applet, such as a voice interface that can recognize and synthesize voice commands, as well as command robots with joysticks or other game devices, as described in the previous section.

Visual

In addition to RoboLab, we use another tool called Visual to teach some Computer Vision aspects [12, 13]. It allows the student quickly to design and test computer vision algorithms, by means of a very friendly user interface based on high-level graphical schemes with a more intuitive use than other commercial applications [30]. The layout of the Visual user-interface is shown in Fig. 5.

A graphical scheme is composed of OPIs (Image Processing Objects). Each OPI represents an operation to do in a scheme; it has a set of inputs that get the images from other OPIs, and it has a set of outputs that return the results after the execution. Each output of an OPI can be connected with one or more inputs of other OPIs by means of pipes that represent the interchange of data or images. At any time, the user can see the images processed in any output or input by clicking with the mouse over it.

In contrast to other similar applications for computing vision [30–32], Visual uses an open modular library of OPIs, and also allows the student to implement new operations, in any

programming language, without worrying about aspects of the graphical user interface. Thus, this tool can also be used on high-level Computer Vision courses.

COURSE OF ROBOTICS AT THE UNIVERSITY OF ALICANTE

We have used the virtual laboratory described in previous section since the 1999/2000 academic year in practice groups of the Robotics course 'Robots and Sensorial Systems'. This is an optional course (see discussion on Spanish university degrees above) in the degrees in Computer Engineering (C.E.), Computer Systems (C.S.) and Management Computing (M.C.) at the University of Alicante. It has at all 7.5 European Credit Transfer System (ECTS), and its program includes 4 hours per week (2 hours theoretical and 2 hours practical) for a period of 4 months for the classroom lessons in addition to other student work.

The course combines Robotics with sensorial techniques including mainly Computer Vision algorithms [33]. A fundamental aspect of Robotics is the use of sensors for the inspection and recognition of objects and elements present in the work setting. Among sensing techniques, Computer Vision is a form of technology that is in constant progress and is very present in current research. The use of Computer Vision in robotic systems allows the computation of the positions of the objects to be manipulated, without the need to make a contact with them.

The optional nature of the topic is an important factor to be considered in the planning of its teaching. We should emphasize that it is the only

subject in the plan of studies in which the basic principles of Robotics are taught, although there is another subject that includes mobile robots programming. Moreover, due to the great number of students registered in this subject, it has become necessary to consider more effective methods for the students' practicals that would allow each student to have access to expensive and limited resources such as robots.

Practical exercises

Considering the above-mentioned aspects and the knowledge that the students gradually acquire when they attend theoretical lessons [33], we propose a set of practical exercises for them. These exercises can be carried out by the student either in the common laboratory for computer science that is assigned to the course during their hours of practical lessons, or remotely, from any other location. In either case, the student has to use the tools of the virtual laboratory to solve the practical exercises. The practical exercises are as follows.

- Exercise 1 (2 hours). Introduction to the tools of the virtual laboratory (RoboLab and Visual), which are used for the different experiments. Study of the components and sub-systems of a robot. Coordinate systems and homogeneous transformations.
- Exercise 2 (6 hours). Kinematics of a robot. Direct kinematics of a 5-DOF robot. Inverse kinematics of a 5-DOF robot.
- Exercise 3 (6 hours). Computer Vision as sensorial technique. 3D Vision. Image processing. Development of a computer vision algorithm. The student resolves this exercise with Visual.
- Exercise 4 (6 hours). Kinematics control of a 5-DOF robot. Manipulation of point-to-point trajectories. Application of a cubic interpolator with parabolic fitting. This exercise is resolved by the student with the aid of RoboLab.
- Exercise 5 (5 hours). Visual servoing. Static see and move approach.
- Exercise 6 (5 hours). Programming of a robot. Modeling of the tasks. Task programming using ACL.

The practical exercises are concatenated in the way that the student first use RoboLab to learn the basic concepts of Robotics. The student then has to calculate the kinematics of a robot (our Scorbot ER-IX). Afterwards, he or she learns to design the appropriate Computer Vision algorithms to detect and locate a certain object in an image using Visual. In addition to the operations that Visual includes, the student is motivated to incorporate new OPIs modules that implement other techniques or algorithms that they consider suitable or necessary. Once the object is located, the kinematics control of the robot is considered for catching the object with the robot. After this problem is resolved with the aid of RoboLab, the visual control techniques are used to do the tasks

of manipulating the object. In the last practical exercise, the student has to program the robot's controller using the information obtained from the previous exercise.

The knowledge assessment of the concepts practices with RoboLab and Visual is based on an automatic correction test via the Internet, as well as the explanations of the results obtained during the exercises.

EXPERIENCES IN TEACHING

We have evaluated the students' opinion, and the effect of the virtual laboratory on learning by means of statistical studies, which have been carried out on fifty students from different practice groups during the 2001/02 and 2002/03 academic years [12]. In this section we describe the main results related to the use of the RoboLab in the last study.

Results related to the use of RoboLab

The first results obtained were that only 20% of the students opted to do the first practical exercise with RoboLab at home in contrast to 17% who preferred to do it at the university laboratory (Fig. 6a). The students who opted to do the practice outside the university (29%) complained of a lag of time and inconvenience. We should point out that, in computer science, many of the students combine study and work. The main reason that the students chose to do their practice at the university's laboratory is that they considered the teacher's explanations to be vital (65% of the questionnaires). Another reason is that they considered the working conditions to be better: the software was already installed and the speed of access was faster. In the second exercise with RoboLab, a considerable reduction is noticed in the number of students who opted to do the exercise at the university's laboratory (57%), as Fig. 6b shows. In

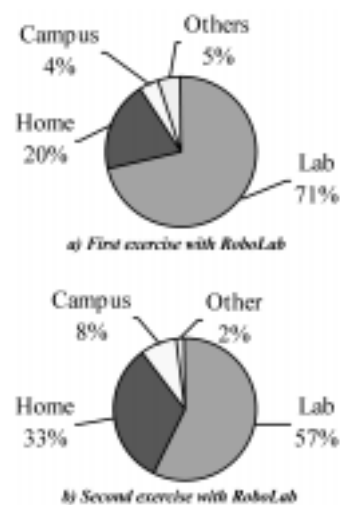


Fig. 6. Student location.

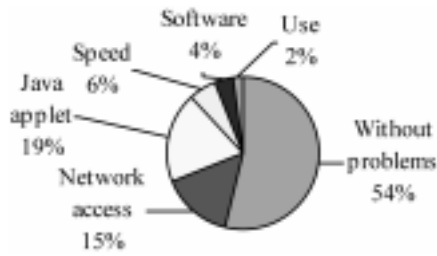


Fig. 7. Student problems with the exercises.

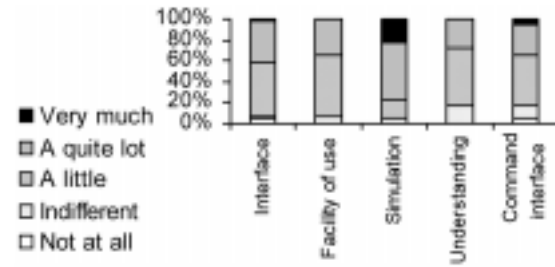


Fig. 9. Appraisal of RoboLab.

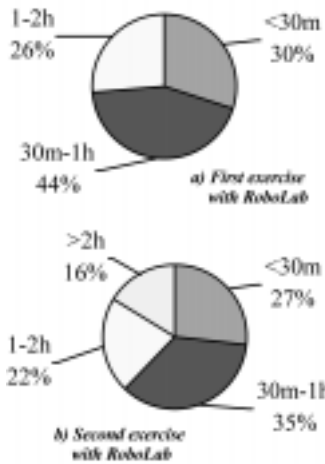


Fig. 8. Time required by the students.

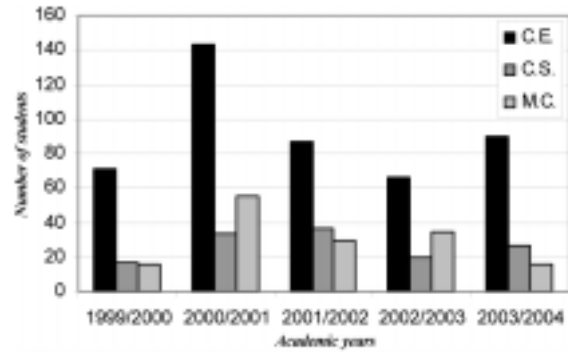


Fig. 10. Change in the number of students registered on the Robotics course.

this case, 33% decided to do their experiments at home.

Figure 7 shows the main problems faced by the students in doing the experiments. 54% did their experiments without any problem. The most common problems were the loading of the applet (19%), due to inappropriate versions of the software, and the access to the Net from outside of the university (15%). Many of the problems have been solved in the version 2 of RoboLab.

An important aspect for evaluating the impact of a virtual laboratory is the facility to assimilate its functioning from the students. Considering that the estimated time of the first practical exercise with RoboLab was 2 hours, and the majority of students did it in 1 hour or less (Fig. 8a), we can conclude that learning with RoboLab is quite simple. For the second practical exercise with RoboLab, 62% of students needed 90 minutes or less, which was the estimated time (Fig. 8b). This gives us an idea of the students' real level of understanding of the use of RoboLab, after been introduced to them in the first exercise.

Figure 9 presents the degree to which the students liked RoboLab. It should be noted that, of the five features evaluated, more than 80% of the students rated it positively, which justifies its inclusion in the virtual laboratory.

Other results

Robotics is a subject that generates great interest from the students; this means that the number of

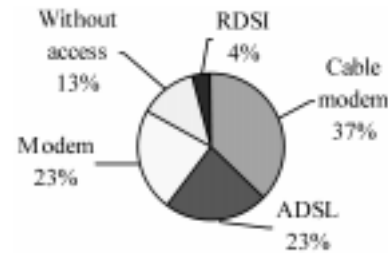


Fig. 11. Internet access used by students.



Fig. 12. Preferences regarding the method of delivery of results of the practicals.

students registered during the last courses has been rising as Fig. 10 shows. Therefore it is necessary to design effective methods for the development of the practicals with the available equipment.

It has been shown that 87% of the students had access to the Internet outside of the university during the 2002/03 academic year (Fig. 11). In the previous year, only 62% had personal access, which showed a considerable increase.

The survey also showed that most of the students preferred to work in the university laboratory, although the majority also preferred to 'hand in' the results of their practicals through the Internet (Fig. 12).

We must point out that the great majority of the students have had no experience with any other type of virtual laboratory with industrial robot simulators that are similar to the one used here, although 25% had previously used some type of mobile robot simulators or had heard about some type of robot arm simulators for commercial use.

CONCLUSIONS

RoboLab is the fundamental tool of our virtual laboratory, and it is primarily designed so that students can practice basics concepts about Robotics and kinematics, and trajectory designing. This tool also allows many students remote access to limited robotic equipment located in a research laboratory, through the Internet, in order to test their results. In addition, the new version of RoboLab offers the student the option of simulating different models of robots with the same user interface, and even operating them remotely if real robots are available. Thanks to RoboLab, students can do their practical exercises by accessing from any location without having to physically go to laboratories or classrooms. In addition, the very friendly RoboLab's user interface makes learning and use by the student very easy, in contrast to other systems.

We use the virtual laboratory in practical exercises for an optional Robotics course at the

University of Alicante, and we have evaluated the student's opinion of it. Based on the studies, an important conclusion can be drawn. Although an increasing number of students happily accept a virtual laboratory that offers them a flexible timetable for their experiments, the majority prefers to have a real laboratory at the university where they can work together with their classmates; they also consider teacher support to be essential. In other words, students consider the virtual laboratory to be a valuable complement to the teacher and traditional teaching, but not as substitute for them. This result contrasts with the current tendency to decentralize education, making it less dependent on the explanations of professors.

In addition, it has been shown that the virtual laboratory is new to our students, although it is well accepted. The simplicity of the virtual laboratory helps the student to save time learning how to use it and gives them more time to concentrate on the more important aspects of the course. The remote access to costly tools and resources like robots is positive and interesting, since they make practice more attractive and real in comparison to a mere simulation.

Finally, it must be stated that the software required for the students' computers is accessible and easy to install, and the tools work well for different types of computers and operating systems. For this, Java is a very good option.

Acknowledgements—We are grateful to the AVCiT of the Generalitat Valenciana regional government for their financial support for the research project CTIDIA/2002/108. We would also like to thank the Education Sciences Institute and the Office of the Vice-President for Studies and Educational Innovation of the University of Alicante for its economic aid.

REFERENCES

1. Ch. Salzmann, P. Saucy, D. Gillet and F. Mondada, Sharing of unique or expensive equipment for research and education, *Informatik / Informatique, Magazine of the Swiss Informatics Societies*, **4**, 1999, pp. 32–33.
2. Ch. Salzmann, H. A. Latchman, D. Gillet and O. D. Crisalle, Virtual laboratories and real-time experiments in engineering education, *Proc. International Conference on Engineering Education (ICEE'99)*, Ostrava-Prague, Czech Republic, (1999).
3. F. A. Candelas, S. T. Puente, F. Torres, F. G. Ortíz, P. Gil and J. Pomares, A virtual laboratory for teaching robotics, *International Journal of Engineering Education*, **19**(3), (2003) pp. 363–370.
4. C. Schmid, A remote laboratory using virtual reality on the Web. *Simulation*, **73**, 1999, pp. 13–21.
5. D. Gillet, Ch. Salzmann, R. Longchamp and D. Bonwin, Telepresence: an opportunity to develop practical experimentation in automatic control education, *European Control Conference*, Brussels, Belgium, (1997).
6. G. T. McKee, An online robot system for projects in robot intelligence, *International Journal of Engineering Education*, **19**(3), 2003, pp. 356–362.
7. R. Marín and P. J. Sanz, Telerobotic training system through the web: low level architecture, *Proc. 1st IFAC Conference on Telematics Applications in Automation and Robotics (TA2001)*, Weingarten, Germany, (2001) pp. 79–84.
8. S. T. Puente, F. Torres, F. G. Ortiz and F. A. Candelas, Remote robot execution through WWW simulation. *15th International Conference on Pattern Recognition (ICPR 2000)*, **4**, (2000) pp. 503–506.
9. J. M. Sebastián, D. García and F. M. Sánchez, Remote-access education based on image acquisition and processing through the Internet, *IEEE Transaction on Education*, **46**, 2003, pp. 142–148.
10. J. Sánchez, F. Morilla and S. Dormido, Tele-operation on an inverted pendulum through the World Wide Web, *Proc. IFAC Workshop Internet Based Control Education (IBCE'01)*, Madrid, Spain, (2001) pp. 37–42.

11. Z. Erlich, J. Gal-Ezer and D. Lupo, *Traditional vs. Technology-Integrated Distance Education. Virtual Environments for Teaching and Learning*, Ch. 2, Series on Innovative Intelligence, Vol. 1, World Scientific, (2002).
12. F. A. Candelas, F. Torres, F. G. Ortiz, P. Gil, J. Pomares and S. T. Puente, Teaching and learning robotics with Internet tele-operation, *2nd International Conference on Multimedia and ICTs in Education (m-ICTE 2003)*, Badajoz, Spain, 3, 2003, pp. 1827–1831.
13. F. Torres, S. T. Puente, F. A. Candelas and J. Pomares, Virtual laboratory for robotics and automation, *Workshop of the IFAC on Internet Based Control Education (IBCE'01)*, Madrid, Spain, 2001, pp. 189–194.
14. Source for Java: <http://java.sun.com>
15. VRML site magazine: <http://www.vrmlsite.com>
16. Java 3D Community site: <http://www.j3d.org>
17. A. Aditya and B. Riyanto, Implementation of Java 3D simulation for Internet telerobotic system, *IASTED International Conference on Modeling and Simulation (MS 2000)*, Pittsburgh, PA, (2000).
18. N. Smith, C. Egert, E. Cuddihy and D. Walters, Implementing virtual robots in Java3D using a subsumption architecture, *Proc. AACE World Conference on the WWW and the Internet (WebNet)*, Honolulu, Hawaii, (1999) pp. 975–980.
19. D. Gillet, F. A. Latchman, Ch. Salzmann and O. D. Crisalle, Hands-on laboratory experiments in flexible and distance learning, *Journal of Engineering Education*, 90, 2001, pp. 187–191.
20. G. T. McKee, An online robot system for projects in robot intelligence, *International Journal of Engineering Education—Special Issue Remote Access/Distance Learning Laboratories*, 19, 2003, pp. 356–362.
21. K. E. Afshari and S. Payandeh, Toward implementation of Java/VRML environment for planning, training and tele-operation of robotic systems, *Proc. 3rd World Multiconference on Systemics, Cybernetics and Informatics (ISAS'99)*, Orlando, (1999).
22. R. Marin, P. J. Sanz and A. P. del Pobil, A predictive interface based on virtual and augmented reality for task specification in a Web telerobotic system, *Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, (2002) pp. 3005–3010.
23. G. C. Burdea, The synergy between virtual reality and Robotics, *IEEE Trans. on Robotics and Automation—Invited Review*, 15, 1999, pp. 400–410.
24. F. A. Candelas, F. Torres, S. Puente, J. Pomares, V. Segarra and J. Navarrete, A flexible JAVA class library for simulating and teleoperating robots, *11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM04)*, Salvador, (2004).
25. K. Goldberg and R. Siegwart, *Beyond Webcams: An Introduction to Online Robots*, The MIT Press, MA, (2002).
26. K. H. Wolf and K. Frotzheim, WebVideo a tool for WWW-based tele-operation, *Proc. IEEE International Symposium on Industrial Electronics (ISIE'97)*, Guimarães, Portugal, (1997) pp. 268–273.
27. F. Torres, S. T. Puente, J. Canovas, J. Mangas and C. Martinez-Larraz, Web tele-operation of robots with simulation feedback, *Telem manipulator and Telepresence Technologies VIII, Proceedings of SPIE*, 4570, (2001) pp. 105–112.
28. H. Sowizral, K. Rushforth and M. Deering, *The Java 3D API Specification*, 2nd edn, Addison-Wesley, Boston, MA, (2000).
29. Jose M. Azorin, Oscar Reinoso, Rafael Aracil and Manuel Ferre, Control of teleoperators with communication time delay through state convergence, *Journal of Robotic Systems*, 21, 2004, pp. 167–182.
30. J. Rasure, R. Jordan and R. Lotufo, Teaching image processing with Khoros, *IEEE International Conference on Image (ICIP-94)*, (1994) pp. 506–510.
31. WiT is an application of Logical Vision: <http://www.logicalvision.com>
32. eVision is a software tool from Euresys: <http://www.euresys.com>
33. F. Torres, J. Pomares, P. Gil, S. T. Puente and R. Aracil, *Robots y Sistemas Sensoriales*, Prentice Hall, Madrid, (2002).

Fernando Torres received his industrial engineer and PhD degrees from the Polytechnic University of Madrid (UPM) in 1991 and 1995 respectively. Since 1994 he has been at the Alicante University, as Professor in control, robotics and computer vision. His research interests include automatic visual inspection, robotics, manufacturing automation, visual servoing, morphological processing and new technologies for teaching. Professor Torres is a member of the CEA-IFAC, IEEE and the Spanish Image Analysis and Pattern Recognition Society. He has published several international papers.

Francisco A. Candelas received his Computer Science Engineer and Ph.D. degrees at the University of Alicante (Alicante-Spain) in 1996 and 2001 respectively. He has been a full-time Professor in the Department of Physics, Systems Engineering and Signal Theory of the University of Alicante since 1999. Since 1996 he has also researched in the Automatics, Robotics and Computer Vision Group in the same department, where he has been involved in several research projects in collaboration with industry and the Spanish Government. His main research interests are virtual and remote laboratories, their application to e-learning, computer vision for real-time applications, and industrial automation.

Santiago T. Puente received his Computer Science Engineer and PhD degrees from University of Alicante in 1998 and 2002 respectively. Since 1999 he has been at the Alicante University, as Professor in robotics and computer networks. Dr. Puente is a member of the CEA-IFAC. His research interests include robotics, manufacturing automation, automatic disassembly, co-operative robots, and new technologies for teaching. Dr. Puente is member of the Advisory Board of the journal *Ventil*.

Jorge Pomares received his Computer Science Engineer and Ph.D. degrees from the University of Alicante, Spain, in 1999 and 2004 respectively. Since 2000 he has been Professor in the Department of Physics, Systems Engineering and Signal Theory of the University of Alicante and teaches courses in robotics and automatic control. Since 1999 he has also researched in the Automatics, Robotics and Computer Vision Group in the same department, where he has been involved in several research projects in collaboration with industry and the Spanish Government. His main research interests are visual servoing, robotics and computer vision. Professor Pomares is a member of the CEA-IFAC.

Pablo Gil Vázquez received his Computer Science Engineer degree from the University of Alicante in 1999. He is currently Assistant Professor in the Department of Physics, Systems Engineering and Signal Theory at the University of Alicante, and teaches courses in computer networks, process control and robotics. His research interests include computer vision problems for robotic systems, image processing and three-dimensional vision, and he is currently working toward the Ph.D. in these topics. He is a member of the CEA-IFAC.

Francisco Ortiz was born in 1974. He received his Computer Science Engineer and the Ph.D. degrees from the University of Alicante (Spain) in 1998 and 2002 respectively. He has been full-time Professor in the Department of Physics, Systems Engineering and Signal Theory of the University of Alicante since 1999. He has been involved in several research projects in collaboration with industry and the Spanish Government. His main research interests are industrial automatization, computer vision, specifically colour systems, mathematical morphology and new technologies for teaching. Professor Ortiz is a member of the CEA-IFAC and has published several international papers.