18th European Symposium on Computer Aided Process Engineering – ESCAPE 18
Bertrand Braunschweig and Xavier Joulia (Editors)

Virtual and Remote Laboratory for Robotics E-Learning

Carlos A. Jara, Francisco A. Candelas, Fernando Torres

University of Alicante, Crta San Vicente del Raspeig s/n, San Vicente del Raspeig, PO Box 03080, Spain.

Abstract

New technologies, such as virtual and remote laboratories, have improved learning and training in the academic community. They allow students to acquire methods, skills and experience related to real equipment in an intuitive and cost-effective way. This paper presents the development of a new e-learning environment in the field of Robotics, which allows students the simulation and teleoperation of a robot arm. The main features of this system, in contrast with others, are its portability, its realistic graphical interface and large amount of options which allow users to learn advanced robotic concepts from home easily. Authors have used *Easy Java Simulations* to develop the software, a Java-based tool intended to create interactive simulations.

Keywords: distance teaching, remote laboratory, telerobotics, virtual laboratory.

1. Introduction

The great evolution of network technologies has allowed the creation of a fast media for global communication between computers: the Internet. This communication channel together with advanced network languages such as Java, were the precursors of *Online Robots* [1]: remote robotic devices which enabled the general public anywhere to provide robotic learning elements. Some of the first successful telerobotic systems controlled through the Internet were the Mercury Project [2], Telegarden Project [3] and the system developed at the University of Western Australia [4], where users could move the robot arm and manipulate objects in the workspace. These Web laboratories opened a new pathway for robotic e-learning.

Because of the explosive growth in the scientific disciplines, as for example robotic systems, students are required to learn and maintain their rapidly expanding knowledge. However, theoretical lessons do not provide enough knowledge to the students. The laboratory work offers them practical exercices to improve their robotic experience. Nevertheless, many problems exist in giving students sufficient educational robotic laboratories. These include expensive equipment, limited time and complicated experimental setups. As a solution to this, the virtual and remote laboratories in telerobotic systems offer a great number of advantages:

- Remote practices and remote access to real equipment.
- Learning in a free and flexible way in contrast to a fixed and regular class schedule.
- Expensive systems can be used.

Therefore, following the essential idea which was mentioned in [5]: "*Educators must have an open attitude towards new technologies,*" most Universities have developed e-learning systems in the educational process. In the case of Control Education, the applications are focused on the study of both stability of simulated systems [6] and the parameter behaviour of real plants [7]. Important in this field are the PEARL project [8], a remote experimental laboratory accessible for students with a range of disabilities, and *iLab* from *iCampus* [9], where there are many remote Web-accessible laboratories developed at Massachusets Institute of Technology. In the field of industrial Robotics, many different applications have been reported since the first telerobotic Web systems mentioned above [2-4]. Among them, it is worth mentioning some *Online Robots* which have appeared with a significant educational purpose:

- ARITI [10]: a telerobotic system that allows to control a robot via the Web with an interface based on *Virtual Reality* (VR) and *Augmented Reality* (AR).
- VISIT [11]: a telerobotic application which has advanced robotic technologies such as computer vision and advanced motion controls to control the robot.
- UJI Robot [12]: a multirobot arquitecture system that gives access to an educational robot arm through the Internet. The system uses AR and VR to manipulate the robot.
- Robolab [13]: an open architecture for simulating and teleoperating different robot arms through the Internet.

Despite the mentioned systems, the virtual and remote laboratory presented here has the advantages of an application based on *Easy Java Simulations*: full portability and an interactive graphical user interface. In addition, it allows one to manage many functions of the robot arm in a realistic way.

The paper is organized as follows. The next section describes the different aspects of the system such as software used, equipment, functional structure and communications. In section 3, the application and his options are showed. Finally, the conclusions and some future work lines are described.

2. Overall System Description

2.1. Easy Java Simulations fundamentals

Easy Java Simulations (Ejs for short) is an open-source tool developed in Java language designed for the creation of interactive virtual labs [14]. The computer simulations created with Ejs can be used as stand-alone Java programs under different operation systems or be distributed via the Internet as applets. The principal reason for choosing Ejs is its simplicity to create graphical simulations. Therefore, there has not been needed a big investment of time to create an interactive graphical user interface. So, it has allowed us to concentrate more on other tasks such as the robotic system model, teleoperation functions and path planning.

2.2. Hardware Components

The hardware description is shown in Fig. 1a. which is based on the Robolab system [13]. There are two well-defined parts separated by the Internet: the user's computer and the laboratory equipment.

The user's computer requires only Internet access, an up-to-date web client, Java and Java-3D runtime components as software. This allows users to use different kinds of computers or operating systems in order to run the application.

In the laboratory, only the robot and its controller require a considerable investment. For the development of the prototype, a Scorbot ER-IX of 5 DOF with an electric gripper has been employed. The main server is a PC that includes the web server from where the user can download the Java application.

The teleoperation server is in another PC that validates the commands that the robot received from a user's computer, translates them to the appropriate language, and sends them to the robot controller. The video server allows users to receive video streams as feedback during the teleoperation. Finally, a computer with Linux is used as a firewall to increase the system's security.

2.3. Software Design

In the system software, there are three main blocks to be considered as shown in Fig. 1b. The client software is an Ejs application. In this program, the main parts are the robot model that manages the 3D simulation based on the library Java-3D, and the functions used in the teleoperation tasks.

When the user gets a command list which has been validated by the simulation, he can request a teleoperation from the application. At this moment, an ASP module (Microsoft Active Server Pages) takes over the control, and verifies the user's identity. If the user is registered in the user database, the ASP module creates a Java module that acts as a communication interface between the client and the teleoperation server.

In the teleoperation server, a Java program attends connections from the main server. A connection includes a command list to be executed in the robot and the corresponding feedback data. Each command list is simulated before its execution in a robot model similar to the client application. This guarantees the correct use of the robot.

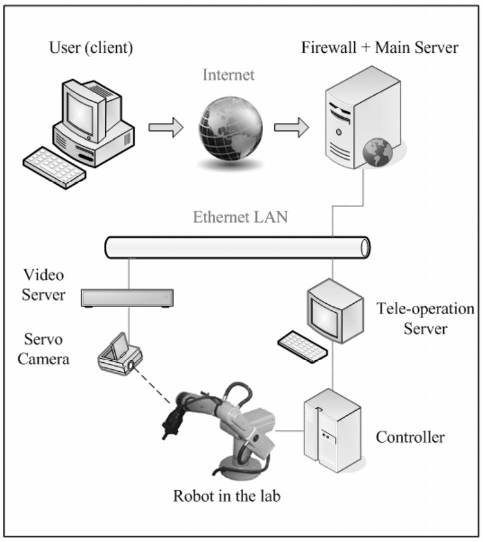


Figure 1a. Hardware components

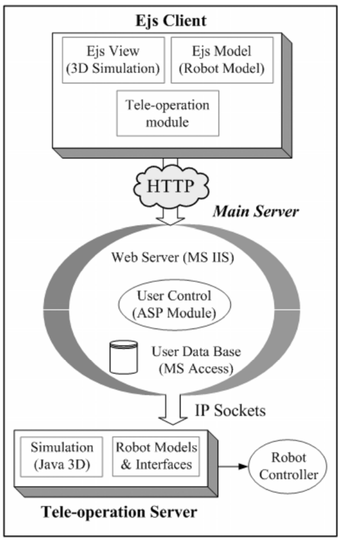


Figure 1b. Software design.

2.4. Communication protocols

The protocol HTTP is used in the communication between the client and the main server. The main advantage of using HTTP is that any connection between a client and the main server is possible, independently of the networks and firewalls to be crossed. Moreover, HTTPS can be used to guarantee the security and privacy of connections.

The data exchanged from the client to the main server is codified as URL strings to be sent in HTTP. These data include information such as the user login, configuration parameters and the command to be executed by the robot arm. On the other hand, the communication from the main server to the teleoperation server is done through TCP sockets. After the client has been connected to the main server, a direct communication between the client and the teleoperation server is established over the HTTP and TCP protocols. This communication also allows feedback data from the server to the client.

The client sends a list of previously-tested commands. Each command list is composed of a type-identifier which represents the order to be executed, the joint values associated with the command, and the times associated to the movements of each joint. The teleoperation server replies HTML data which contains current articular positions of the robot movement.

3. Results

The system developed is accesible in applet form from the following web: http://www.aurova.ua.es/robolab/index.html.

3.1. User interface

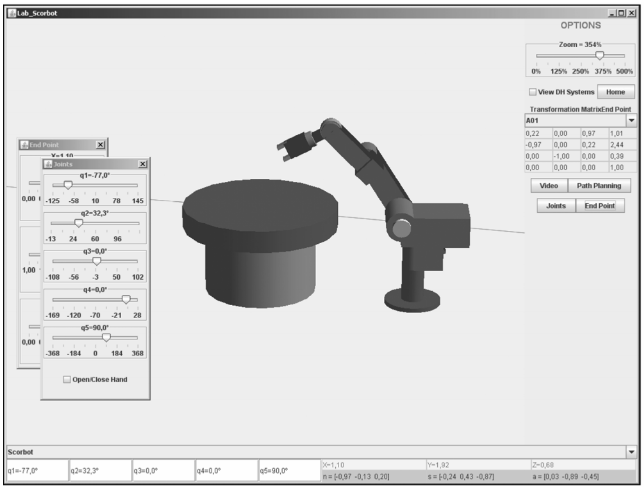
The appearance of the user interface is shown in Fig. 2a. The upper part shows a 3D representation of the workspace where the robot arm is displayed. At the left of the application, there are some control panels where users can specify the exact joint values (direct cinematic) and the Cartesian coordinates of the end effector (inverse cinematic). Finally, on the right, there are several controls with more options such as zoom, transformation matrix of all the links, video feedback, graphical representation and path planning (see 3.2).

3.2. Path planning

Users can practice and carry out movements of five kinds of trajectories: axis to axis, simultaneous axis, synchronous, asynchronous and 4-3-4 polynomial trajectory (see 3.2.1). The simulated trajectories can be stored in a command list and simulate sequentially. The user can also import and export trajectories to the software from a text file easily.

When the user activates the teleoperation option, the command list is sent to the laboratory equipment to be executed by the robot. Before this, the user must configure some teleoperation settings, such as the user's name and password, the URL of the main server and port of the teleoperation server. The interface gives the user two options for performing the feedback of a teleoperation (Fig. 2b):

- An on-line video stream.
- Graphical updating of the 3D simulation with the current position of the real robot.



A01

0,22	0,00	0,97	1,01
-0,97	0,00	0,22	2,44
0,00	-1,00	0,00	0,39
0,00	0,00	0,00	1,00

☐ View DH Systems

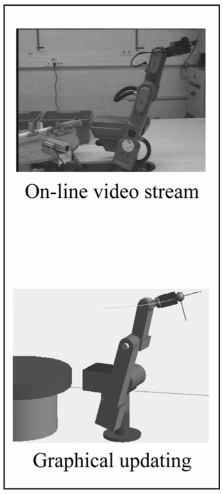
q1=-77,0°
q2=32,3°
q3=0,0°
q4=0,0°
q5=90,0°

☐ Open/Close Hand

Scorbot

q1=-77,0°	q2=32,3°	q3=0,0°	q4=0,0°	q5=90,0°	X=1,10 n = [-0,97 -0,13 0,20]	Y=1,92 s = [-0,24 0,43 -0,87]	Z=0,68 a = [0,03 -0,89 -0,45]

Figure 2a. User interface


On-line video stream

Graphical updating

Figure 2b. Feedback options.

3.2.1. 4-3-4 polynomial trajectories

The model of the system implements the formulation of 4-3-4 polynomial trajectory [15], a smooth robot movement with acceleration and speed continuity. This path planning divides the robot trajectory in three parts: actuator acceleration (a fourth-order equation), time of maximun velocity (a third-order equation) and actuator decelaration (a fourth-order equation too). This type of trajectory is the same that the real robot uses.

In order to validate the path planning model, the simulated and real data of a trajectory were compared. The final experimental results are shown in Fig. 3a, where the similarity between the model and the real system can be seen. Therefore, users can simulate a real robot trajectory and view its temporal evolution.

3.3. Off-line programming

Users can programme routines in the simulation (Fig. 3b). They can create variables, operations and order movements. The trajectories simulated in the routines are stored in the command list to simulate sequentially or to teleoperate.

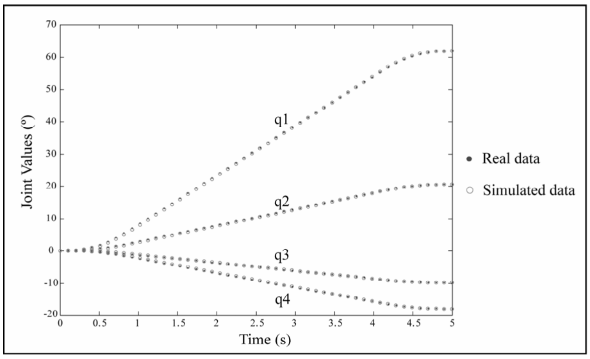
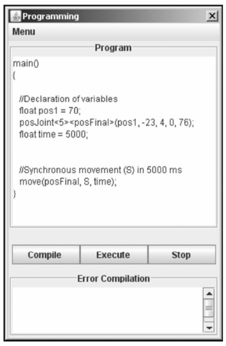


Figure 3a. Simulated and real trajectory



```
main()
{

  //Declaration of variables
  float pos1 = 70;
  posJoint<5><posFinal>(pos1, -23, 4, 0, 76);
  float time = 5000;


  //Synchronous movement (S) in 5000 ms
  move(posFinal, S, time);
}
```

Figure 3b. Interface to program

4. Conclusions and future work

The virtual and remote laboratory presented is mainly oriented to the training and e-learning in robotics. This application has been used in teaching in the course “Robots and Sensorial System” in the Computer Science Engineering degree. With this sytem, students can learn robotic concepts such as direct/inverse cinematic, path planning, teleoperation and programming. One of the main and distinctive features of this application is the great amount of trajectories to simulate and to teleoperate. Because the program is an applet, users can run it from other places in free schedules.
As for future works, the system can be improved with new paths such as adding new robots, 3D object recognition and inclusion of virtual objects using augmented reality.

5. Acknowlegment

The work presented is supported by the Spanish Government through FPI grants program and research project DPI2005-06222.

References

[1] K. Goldberg and R. Siegwart, Beyond Webcams: an introduction to online robots, MIT Press, Cambridge, 2001, pp.61-98.

[2] K. Goldberg, S. Gentner, C. Sutter and J. Wiegley, The Mercury Project: a feasibility study for Internet robots, IEEE Robotics & Automation Magazine, 7 (2000) 35.

[3] K. Goldberg, M. Kusahara, H. Dreyfus, A. Goldman, O. Grau, M. Gržinic, B. Hannaford, M. Idinopulos, M. Jay and E. Kac, The robot in the garden: telerobotics and telepistemology in the age of the Internet, MIT Press, Cambridge, 2000.

[4] K. Taylor and B. Dalton, Internet robots: a new robotics niche, IEEE Robotics & Automation Magazine, 7 (2000) 27.

[5] S. Dormido, Control Learning: Present and Future, Annual Reviews in Control, 28 (2004) 115.

[6] S. Dormido, F. Gordillo, S. Dormido-Canto and J. Aracil, An interactive tool for introductory nonlinear control systems education, 15th IFAC World Congress, Barcelona, 2002.

[7] J.S. Fujita, R.F. Cassaniga and F.J. Fernandez, Remote Laboratory, Proc. IEEE Int. Symposium on Industrial Electronics, 2 (2003) 1104.

[8] C. Colwell, E. Scanlon and M. Cooper, Using remote laboratories to extend access to science and engineering, Computers & Education, 38 (2002) 65.

[9] S. Lerman and J. Del Alamo, iLab: Remote Online Laboratories, [Online], avalible: http://icampus.mit.edu/projects/iLabs.shtml.

[10] Augmented Reality Interface for Telerobot Application Via Internet, [Online], avalible: http://lsc.cemif.univ-every.fr:8080/Projects/ARITI/index.html.

[11] K. Kosuge, J. Kikuchi and K. Takeo, VISIT: A teleoperation system via the computer network, Beyond Webcams:an introduction to online robots, MIT Press, Cambridge, 2002.

[12] R. Marín, P.J. Sanz, P. Nebot and R. Wirz, A Multimodal Interface to Control a Robot Arm via the Web: A Case Study on Remote Programming, IEEE Transactions.on Industrial Electronics, 52 (2005) 1506.

[13] F. A., Candelas, C.A. Jara and F. Torres, Flexible virtual and remote laboratory for teaching Robotics, 4th Int. Conf. on Multimedia and Information & Communication Technologies in Education, 3 (2006) 1959.

[14] F. Esquembre, Easy Java Simulations: a software tool to create scientific simulations in Java, Computer Physics Communications, 156 (2004) 199.

[15] K.S. Fu, Robotics: control, sensing, vision and intelligent, Mc Graw Hill, 1987, pp. 160-168.