RobUaLab.ejs: a new tool for robotics e-learning

C.A. Jara, F.A. Candelas and F. Torres
Department of Physics, System Engineering and Signal Theory, University of Alicante, Spain

Abstract— 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. The purpose of this paper is to present the development and the implementation of an e-learning environment in the field of Robotics. This application allows students to simulate and to teleoperate a robot arm in an easy and user-friendly way. The application has been developed using Easy Java Simulations (EJS), an open-source tool for people who do not need complex programming skills.

Index Terms— Educational technology, simulation software, telerobotics, virtual and remote laboratories.

I. INTRODUCTION

Theoretical lessons do not provide enough robotic knowledge for students. Laboratory work offers them practical issues to improve their robotic experience. However, many problems exist in giving students sufficient educational robotic laboratories. These include expensive equipment and limited time. As a solution to this, new technologies such as Virtual and Remote Laboratories in telerobotic systems provide a distance teaching framework which meets the student hands-on learning needs. They solve the problems mentioned above, offering a great number of advantages such as remote practices and remote access to real equipment, learning in a free and flexible way and the use of expensive equipment.

This paper presents a virtual and remote laboratory called RobUaLab.ejs for training and learning in robotics. In general, this application allows a complete simulation of a robot arm by means of a virtual environment based in a Java applet. Path planning algorithms validated in the simulation can be executed in a real remote robot through the Internet. In addition, this telerobotic system developed has other interesting features such as: 1) feedback to the user both on online video and graphical updating of the 3D simulation; 2) the use of high level protocols (HTTP/HTTPS) to have a transparent communication; and 3) a very realistic graphical interface (Fig. 1).

Nowadays, there are other robotic e-learning systems developed for an educational purpose. Among them, it is worth mentioning the following:

- ARITI [1]: a telerobotic system that allows to control a robot with an interface based on Virtual Reality (VR) and Augmented Reality (AR).
- UJI Robot [2]: a multi-robot architecture system that gives access to a robot arm by means of an interface which uses VR and AR.
- Robolab [3]: an open architecture for simulating and tele-operating different robot arms through the Internet.

In contrast to the above mentioned systems, the virtual and remote laboratory presented here has the advantages of an application based on Easy Java Simulations (EJS) [4]: full portability and an interactive graphical user interface based on VR and AR. In addition, it allows one to manage many robotic functions which are novel in a free Java application like this.

The e-learning application presented is being used in the course “Robot and Sensorial Systems” in the Computer Science degree at the University of Alicante. Moreover, this system belongs to a network performed by different virtual and remote laboratories from Spain, coordinated by Prof. Sebastian Dormido called “AutomatLab” (http://lab.dia.uned.es/automatlab). With this virtual platform, students and teachers can experiment with real equipment and share knowledge by means of a collaborative environment based in eMersion [5].

This paper is organized as follows. Section 2 discusses the technical implementation of this robotic system, which includes the hardware components, software architecture and communication protocols. Afterwards, Section 3 describes the main features of the virtual laboratory developed. The remote and teleoperation capabilities are shown in Section 4. Finally, the conclusions and some future work lines are shown in Section 5.

II. TECHNICAL IMPLEMENTATION

A. Hardware Components.

The different hardware components are shown in Figure 2. There are two clear parts linked by the Internet: the user’s computer and the laboratory equipment. The User’s PC requires only Internet access, a web browser, Java and Java 3D runtimes installed. This allows users to use different kinds of computers or operating systems in order to run the application.

In the laboratory, the only pieces that require a considerable investment are the robot arm and its controller. For the development of this robotic lab, a Scorbot ER-IX (Intellitek) of 5 DOF with an electric gripper is used for teleoperation.
The Main Server is a PC which includes the web site from where the user can download the Java application to simulate and to teleoperate the robot. The Tele-operation Server validates the commands that the robot receives from a user’s computer, translates them to the appropriate language, and sends them to the robot controller. The IP camera allows users to receive video streams by means of the HTTP protocol as feedback during the teleoperation processes. Finally, a PLC permits remote power control of the laboratory.

B. Software architecture.

With regard to the software design, there are three main blocks to be considered: the Client Applet, the Main Server and the Tele-operation Server software (Fig. 3). The client software is an EJS application which contains the robot model that manages the 3D simulation (based on the Java 3D library), and the functions used in the teleoperation tasks.

When the user gets a path planning validated by the simulation, he can request the teleoperation module from the application. At this moment, a PHP module takes over the control, and verifies the user’s identity. If the user is registered in the user database, the PHP module creates a socket communication which acts as a bridge between the Client Applet and the Tele-operation Server.

The Tele-operation Server is a Java program that attends connections from the Main Server. A connection includes a command list to be executed in the robot and the corresponding feedback data. When the Tele-operation Server receives a command list, it does a simulation of the commands in order to verify that they are correct. This simulation is based on the same robot model as the client application and guarantees the correct use of the robot.

C. Communication protocols

The protocols HTTP and HTTPS are used in the communication between the Client Applet and the Main Server (Fig. 3). The main advantage of using high-level protocols is that any connection between a client and the web server is possible, independently of the networks and firewalls to be crossed, provided that the necessary ports are enabled. This simplifies the use of the application since users do not have to configure any network device.

The data exchanged between the client and the Main Server is codified as URL strings to be sent in HTTPS. These data include information such as the user login, configuration parameters or the command to be executed by the robot arm. On the other hand, the communication between the Main Server and the Tele-operation Server is done through TCP sockets because the computers are in the same private LAN. After the client has been connected to the Main Server and the login authentication process has been successful, a direct communication between the client and the teleoperation server is established over the HTTPS and TCP sockets protocols. This communication allows the exchange of commands from the client to the Tele-operation Server and feedback data in the opposite direction.

The client sends a list of commands, which are previously tested in a simulation. Each command of the list is composed of a type-identifier which represents the order to be executed, the joint values associated with the movements of each joint. It should be pointed out that the system does not perform a control-loop of the robot through the Internet. Only tested command lists are sent to the Tele-operation Server to be executed remotely by the robot arm, in order to get real movements.

Finally, the HTTP protocol is also used to remote control both the PLC and the IP camera (Fig. 3).

III. THE VIRTUAL LABORATORY

The virtual lab developed implements a large amount of options suitable for robotic e-learning. Students will be able to learn complex robotic concepts by means of a VR environment in an easy way. This section describes the main features of the virtual part of the applet and all the possibilities which are implemented for the user experimentation.
A. User interface

The appearance of the user interface is shown in Figure 4. The lower part on the left shows a 3D representation of the workspace where the robot arm is displayed. This robotic simulation has been developed using the Java 3D capabilities of EJS and represents a complete virtual model of the real environment. On the right of the application, there are some control display panels where users can view the time evolution of some model variables: position and speed (Pos_Speed panel), acceleration and actuator torque (Dynamics panel), transformation and Jacobian matrices (DataC panel), and the dynamic equation matrices (DataD panel). The control menu located in the upper part of the diagram allows users to save the experiments performed both in image format and in Matlab m-file format (eJournal option). This permits users to share experiment results with others by means of the collaborative environment where the application is embedded. Finally, the upper part of the left contains several button controls with options which will be explained in the next subsection.

B. Virtual laboratory’s options

The virtual environment developed allows users to experiment with a lot of options. Many of them are novel in a free Java application like this. Among them, it is worth pointing out:

- Cinematics: users can move the robot specifying both the exact joint values (direct cinematic) and the cartesian coordinates of the end effector (inverse cinematic). Denavit-Hartenberg systems and transformation matrices can be seen in the user interface.

- Path Planning: users can practice and carry out movements of both joint trajectories (synchronous, asynchronous, splines and 4-3-4 polynomial trajectory) and cartesian trajectories (line). The simulated trajectories can be stored in a command list and simulated sequentially. The user can also import and export trajectories to the software from a text file easily (Fig. 5).

- Environment modeling: users can introduce virtual objects in the workspace to do pick & place operations.

- Dynamics: users can evaluate the torques in the actuators when the virtual robot is simulating a task. They can modify dynamic parameters such as link masses, inertias and viscous friction from the robot and realize how the dynamics change.
• **Programming**: users can programme Java routines in the simulation. They can create variables, mathematical operations and order movements. The trajectories simulated in the routines are stored in the command list to simulate sequentially.

• **Virtual Camera**: users can view a virtual workspace projection of an eye-in-hand virtual camera. This option will be used to perform visual servoing applications in future developments.

• **Visualization data**: as mentioned before, users can view in real-time all the values about the position transformation, cinematic and dynamic models of the virtual robot (Fig. 6).

### IV. REMOTE CAPABILITIES

The application presented allows to control remotely real equipment through the Internet. These remote experiences enhance the accessibility of experimental setups providing a distance teaching framework which meets the students hands-on learning needs. The next subsections explain the remote capabilities of the system and the way to access them.

#### A. Schedule system

The application is embedded in a restricted environment. Authorized students can download the applet at anytime from anywhere and experiment with only the virtual laboratory. Remote access to real equipment is controlled by a schedule system. Thus, users can make a reservation of the real lab specifying the experiment timetable (day and time). This action creates a new line in the **User Data Base** (Fig. 3) with the user and experiment data (name, password, start and end of time of remote access). A thread process installed in the web server checks that users are in their correct timetable when they experiment with the real lab. In this way, only one user can control the robotic plant at the same time.

#### B. Telesoperation options

The application allows the execution of high level tasks permitting users to interact with the real plant in a friendly and easy way. This remote experimentation is based on the high level protocols HTTP/HTTPS. This way, users do not have to open any port for the teleoperation and they only need a common Internet connection.

The teleoperation options implemented in the application allows the remote control not only of the robot, but also of some electronic devices of the real laboratory. They are the following:

• **Remote PLC/Camera control**: authorized users can control from the applet both some PLC control parameters (switch on/off both the light and the robot controller) and the real camera projection (pan, tilt and zoom).

• **Remote robot control**: according to a schedule, users are able to execute remotely in the real robot the command list stored in the virtual simulation. As mentioned before, the path planning sent to the real robot is previously checked in the **Teleoperation Server** which detects the possible collisions of the robot-arm with its environments and with itself.

• **Feedback options**: the application gives the user two options for performing the feedback of a teleoperation: an online video stream and graphical updating of the 3D simulation with the current position of the real robot.

• **Augmented Reality**: the real information from the robot scenario is complemented with some virtually generated data from the virtual environment (Fig. 7). Virtual projection is combined with the current state from the remote laboratory. This feature helps to improve user performance and provides more information to control the robot.
V. CONCLUSIONS AND FUTURE WORK

In this paper, a virtual and remote laboratory for the simulation and teleoperation of an industrial robot arm has been presented. Our system is mainly oriented towards the training and e-learning of robotic concepts.

The application has been developed using EJS, an open-source tool designed for the creation of interactive simulations. In this way, the procedure to transform the robotic system in an interactive virtual laboratory has been easier to do than the majority of programs available. It has not been necessary to learn specific programming skills and a big investment of time has not been needed to create the application.

With the virtual lab developed, students can learn robotic concepts such as direct/inverse cinematic, path planning, dynamics and programming. The user interface is very user-friendly, and the graphical simulation very realistic.

The remote capabilities of the application allows users to experiment with real equipment through the Internet. Remote experimentation of high level tasks based on AR encourages students to learn robotic concepts and provides them with a realistic hand-on experience.

Finally, the system presented collects a lot of interesting virtual and remote features (complete robot simulation, robot dynamics, remote power and robot control, augmented reality, etc.), which are difficult to find together in a free Java applet like this.

At present, authors continue developing the application described here and are working on including new features such as 3D recognition of basic objects in the workspace and visual servoing algorithms.

ACKNOWLEDGMENT

This work was supported in part by the “Ministerio de Educación y Ciencia” of the Spanish Government through FPI grants program.

REFERENCES


AUTHORS

Carlos A. Jara is with the University of Alicante, Spain. Email: cajbdfists.ua.es.
Francisco A. Candelas is with the University of Alicante, Spain. Email: francisco.candelas@ua.es.
Fernando Torres is with the University of Alicante, Spain. Email: fernando.torres@ua.es.

This work was supported in part by the research project DPI2005-0622.