An augmented reality interface for training robotics through the web

Carlos A. Jara, Francisco A. Candelas, Pablo Gil, Manuel Fernández and Fernando Torres

Department of Physics, System Engineering and Signal Theory, University of Alicante
03690, San Vicente del Raspeig, Spain (emails: {carlos.jara, francesco.candelas, pgil, fernando.torres}@ua.es)

Abstract: The exponential spread of the World Wide Web has had an enormous impact on educational institutions. New technologies, such as virtual and remote laboratories, have improved learning and training in the academic community, mainly in the field of technical and science subjects. 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 last improvements of an e-learning environment, based on a Java applet, in the field of Robotics. This application allows students the simulation and teleoperation of a robotic arm by means of an interface with augmented reality support. Other main features of this system are its realistic graphical interface and its portability, which have been possible thanks to Easy Java Simulations, a Java-based tool which has been used to create this application.

1. INTRODUCTION

The great evolution of network technologies has allowed the creation of the Internet. This communication channel together with advanced network languages such as Java were the precursors of Online Robots (Goldberg and Siegwart, 2002): remote robotic devices which enabled people to provide adequate robotic learning elements. Some of the first successful telerobotic systems controlled through the Internet were the USC Mercury Project (Goldberg et al., 2000a) and the Telegarden project (Goldberg et al., 2000b), where users could move the robot arm and manipulate objects in the workspace. These Internet-based remote laboratories opened a new pathway for robotics e-learning.

Because of the explosive growth in the robotics discipline, as for example robotic systems, students are required to maintain their rapidly expanding knowledge. However, theoretical lessons do not provide enough knowledge to students. Laboratory work offers them practical issues to improve their robotic experience. Nevertheless, many problems such as expensive equipment and limited time, prevent teachers having sufficient educational robotic laboratories. As a solution to this, new technologies such as Virtual Reality (VR) or Virtual and Remote Laboratories (VRL) offer a great number of advantages such as remote practices and learning in a free and flexible way.

Therefore, following the essential idea which was mentioned in (Dormido, 2004): “Educators must have an open attitude towards new technologies,” a virtual and remote laboratory called RobUaLab.ejs (http://robuulab.epcs.ua.es) has been developed for training and learning in robotics. This system allows users to simulate and test positioning commands for a robot by means of a virtual environment with augmented reality (AR) support, as well as execute the validated commands in a real remote robot through the Internet. The application has been developed using Easy Java Simulations (EJS) (Esquembre, 2004) which have allowed a full portability and an interactive graphical user interface based on VR and AR. Moreover, it is important to point out other interesting features of the application, such as: 1) feedback to the user while the robot is moving by means of an online video stream and graphical updating of the 3D simulation; 2) the use of high-level communication protocols (HTTP/HTTPS); and 3) a very realistic user interface. In addition, this e-learning application presented is being used in the course “Robot and Sensorial Systems” at the University of Alicante and belongs to a network of different virtual and remote laboratories. This project is called “AutomatL@bs” (http://lab.dia.une.edu/automatlab) and it allows users to share knowledge by means of a collaborative environment based in eMersion (Gillet et al., 2005).

The paper is organized as follows. First, a brief state of the art about VRL is explained. Afterwards, Section 3 describes the different aspects of the system. Section 4 explains the main features of the application developed. Next, some experiments performed are explained in Section 5. Finally, some important conclusions are shown in Section 6.

2. RELATED WORK

Nowadays, many fields of engineering use e-learning systems for education. In the case of Control Education, some applications are focused in the study of stability of linear and nonlinear systems (Dormido et al., 2002). Important in this field is the PEARL project (Colwell et al., 2002), a remote laboratory accessible for students with a range of disabilities. Some more examples of e-learning platforms in other engineering fields, such as Image Processing and Industrial Automation, can be seen in the following references (Sebastian et al., 2003, Candelas and Moreno, 2005).

In the field of industrial Robotics, many different systems have been reported since the USC Mercury Project and the Telegarden project. However, there are fewer applications with educational purpose. Among them, it is worth mentioning the following:
1. VISIT (Kosuge et al., 2002): a telerobotic application which has advanced robotic technologies.

2. UJI Robot (Marin et al., 2005): a multi-robot architecture system that gives access to a robot arm through the Internet. The system uses AR and VR to manipulate the robot.

3. Robolab (Candelas et al., 2003): an open architecture for simulating and teleoperating different robot arms.

Table 1 shows a comparative study among the telerobotic systems mentioned above and the presented in this paper (RobUaLab).

### Table 1. Comparative study among VRL

<table>
<thead>
<tr>
<th>Property</th>
<th>VISIT</th>
<th>Robolab</th>
<th>UJI</th>
<th>RobUaLab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web based</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Virtual Reality</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Augmented Reality</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>High Level Commands</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Programming</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Booking System</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Dynamics model</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Object Recognition</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

As a conclusion of this comparison, the application presented allows users to manage many robotic functions which have not been implemented before in a free Java application like this.

3. OVERALL SYSTEM DESCRIPTION

3.1 Hardware components

The different hardware components are shown in Figure 1. 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 components as software. This allows users to use different kinds of computers or operating systems in order to run the application.

In the laboratory, there are three main components: the robot arm, its controller and the automatic conveyor belt. For the development of this robotic lab, a robot Scorbot ER-IX of 5 DOF with an electric gripper is used for teleoperation.

The Main Server is a PC which includes the web site from where users can download the Java applet. The Tele-operation Server validates the commands that the robot receives from a user’s computer, translates them to the appropriate robot 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 connected with the Main Server permits remote power control of the real laboratory.

3.2 Software design

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. 2). The Client Applet is an EJS application that can be downloaded from the Main Server. In this program, the main parts are the robot model which manages the 3D simulation, based on the EJS’ 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 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 applet and guarantees the correct use of the robot.

3.3 Communication protocols

The protocols HTTP and HTTPS are used in the communication between the Client Applet and the Main Server (Fig. 2). 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. This simplifies the use of the application since users do not have to configure any network device or firewall. In addition, HTTPS protocol provides a secure method for transferring data through the Internet.

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 and the command to be executed by the robot arm.

![Fig. 1. Hardware components](image-url)
On the other hand, the communication between the Main Server and the Tele-operation Server is done through TCP sockets and UDP packets. After the client has been connected to the Main Server and the login authentication process has been successful, a communication channel between the Client Applet and the Teleoperation Server is established over the HTTPS and TCP/UDP protocols. This communication allows the exchange of high level commands from the client applet to the Teleoperation Server and feedback data in the opposite direction.

3.4 Augmented reality issues

As commented in the Introduction, the application implements an interface based on AR. This technology provides users an environment which combines computer 3D graphics together with images from the remote environment. In order to achieve an interface with AR support, IP camera images obtained through the HTTP protocol, are placed into the background of the application panel (Fig. 4). The virtual world which represents the real workspace, is put on the same panel. Virtual camera projection that visualizes the virtual environment is configured with the same extrinsic parameters that the real camera has. In this way, the virtual world matches with the real world giving more information to control the real robot.

4. USER INTERFACE DESCRIPTION

4.1 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 user experimentation.

The appearance of the user interface is shown in Figure 3. The lower part on the left shows a 3D representation of the workspace. This robotic simulation has been developed using Java 3D capabilities 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 equation matrices for dynamics (DataD panel). The control menu located in the upper part 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 other users by means of the collaborative environment.

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:

1. Kinematics: users can move the robot specifying both the exact joint values (direct kinematics) and the cartesian coordinates of the end effector (inverse kinematics). Denavit-Hartenberg systems, transformation and Jacobian matrices can be seen in the user interface. In addition, the application detects possible singularities in the robot.

2. 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. Trajectories can be also imported from and exported to text files easily by the user.

3. Environment modelling: users can introduce specific virtual objects to do pick & place operations.

4. 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 and inertias of the robot and realize how the dynamics change.

5. Off-line-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. Besides, users can import and export programs from a text file.


4.2 Remote interaction

The application presented allows controlling remotely real equipment through the Internet. These remote experiences enhance the accessibility of experimental setups providing a distance teaching framework which meets students hands-on learning needs. Currently, remote operation is only available when students access to the AutomatL@bs project web site as a registered user.
The application is embedded in a user restricted environment. Authorized students can download the applet at anytime from anywhere and experiment with only the options of the virtual laboratory. Remote access is controlled by a schedule system included in the AutomatL@bs’ environment. Thus, users can make a reservation of the real lab specifying the experiment timetable. This action creates a new line in the User Database from the Main Server (Fig. 2) with the user and experiment data (name, password, start and end of time of remote access). A thread process installed in the Main 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 and it avoids multiple user connections.

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 and HTTPS. This way, users do not have to open any port or firewall for the teleoperation and they only need a common Internet connection.

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

1. Remote PLC/Camera control: users can control from the applet both some PLC control parameters (switch on/off operations) and the real camera projection.

2. 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.

3. 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.

4. Augmented Reality: the real information from the robot scenario is complemented with some virtually generated data from the virtual environment (Fig. 4). Virtual projection is combined with the current state from the remote laboratory taking into account current IP camera setting and the 3D environment. This feature helps to improve user performance and provides more information to control the robot.

4.3 Object recognition algorithm

The Java applet contains a programming module to recognize basic objects from the IP camera images. In this way, objects from the real plant can be imported to the virtual laboratory.

Figure 5 illustrates the algorithm implemented. Basically, the input image is smoothed by an operator to eliminate Gaussian noise. Afterwards two types of segmentation process are particularly applied: segmentation based on RGB colour model and HSV colour model. For RGB segmentation, the chosen criteria is to compare clusters’ variation using the Euclidean distance between the three component colour and reference values for each cluster.

On the other hand, for HSV segmentation, each pixel from image is only labelled considering the hue and saturation values. Finally, the block “Extract Colour” shows an image where only will be represented the objects with the colour proposed. From this last image, the algorithm computes the position and orientation for all objects of this colour. This allows the application to do grasping tasks by means of the teleoperation of the remote robot using the virtual laboratory.
5. CASES OF STUDY

5.1 Pick and place

This programming experiment consists of doing a pick-and-place operation of an object located in the conveyor belt. Figure 6 shows the corresponding program of this task. There are three different parts in the program’s structure: 1) declarations of the positions; 2) creation of the objects \( posJ \) (joint position); 3) definition of the order movements. In these last methods, users can specify the trajectory that they want to use. In the example proposed, the task is performed by means of a 4-3-4 polynomial trajectory (parameter “434” in \( moveJ \) method), a smooth robot movement with acceleration and speed continuity.

Figure 6 shows the states of the virtual robot during the execution of the pick-and-place experiment. The image sequence represents each of the joint positions programmed in the routine. Before starting the virtual robot, the conveyor belt is switched on to detect the object. This order correspond with the method “belt()” of the Java program (Fig. 6).

5.2 Remote experiment using the AR interface

This subsection shows the remote execution of a path planning experiment. As discussed before, the teleoperation is based on high level movement commands over the HTTPS protocol. Thus, the trajectories validated in the simulation will be sent for remote execution in the real plant. In order to ensure the correct use of the robot, the application applies a security system based on the following criteria:

1. The Robot Model of the Client Applet (Fig. 2) checks that the trajectory simulated does not exceed the maximum velocity allowed in the robot. In addition, the virtual environment validates that the trajectory simulated has not any singularity.

2. The Teleoperation Server software checks that the trajectory sent to the robot does not have any collision with other objects from the workspace. This program uses the same model that the virtual laboratory implements.

The experiment proposed consists of a synchronous trajectory of four seconds long. Figure 7 shows the states of the real robot together with the user interface during the execution of the path planning experiment. The right image represents the IP camera video stream, and the left image shows the graphical updating of the 3D simulation of the AR interface.

As it can be seen in the image sequence, 3D graphical updating is slightly delayed regarding the real image. This fact is due to feedback data from the current position of the real robot is performed through HTTPS requests and this protocol is always put down at some delay time. Despite this drawback, the communication system of the application was able to receive ten different joint coordinate values from the real robot for a trajectory of four seconds long. This involves a mean rate about three update values per second.
6. CONCLUSIONS

In this paper, a virtual and remote laboratory for the simulation and teleoperation of an industrial robot arm, has been presented. This 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.

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 allow users to experiment with real equipment. Remote experimentation of high level tasks based on AR encourages students to learn robotic concepts and provides them with realistic hands-on experience. The communication between the user and the Main Server is based on HTTPS protocol which offers the advantages of an easy configuration and a security interaction. Despite of this, this protocol causes some delay time in the 3D graphical feedback.

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. The advanced features which contains the application only are usually available in professional or specific software tools.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by the “Ministerio de Educación y Ciencia” of the Spanish Government through research project DPI2005-06222.

REFERENCES


