

# The REAL Lab: Remote Experiments for Active Learning

M. Cefalo, L. Lanari, G. Oriolo, M. Vendittelli

*Dipartimento di Informatica e Sistemistica  
Università degli Studi di Roma "La Sapienza"  
Via Eudossiana 18, 00184 Roma, Italia  
E-mail: {cefalo,lanari,oriolo,vendittelli}@dis.uniroma1.it*

**Abstract:** This paper describes the realization of a remote laboratory called REAL (Remote Experiments for Active Learning) Lab. Mainly designed for educational purposes, the REAL Lab allows the execution of robotic control experiments through the Internet. The client-server architecture of the telelaboratory is based on the Java technology, making the user interface platform-independent; furthermore, no installation of commercial software is required on the client side. When connected to the Lab, the user must choose between two modes of operation, i.e., performing virtual or physical experiments. In both cases, the behavior of the controlled robotic system is shown to the user in some visual form, i.e., through plots, videos or animations. The robotic experiments currently available at the REAL Lab have been selected to highlight control problems of educational and research interest.

## 1 Introduction

The role of computer technology in education is increasing at a tremendous rate, and many advanced teaching tools are nowadays Internet-based. The effectiveness of web-based active learning is currently under investigation: empirical results clearly show that interactive learning environments are superior to passive learning environments in terms of the learning outcome [1]. The combination of the Internet technologies with "learning by doing" strategies in the educational processes is expected to deeply modify the teaching methods and the notion of education itself [2].

Despite the potential of the new technologies, many researchers are still struggling to identify the correct perspective for taking full advantage of the Internet in educational processes. A key concept for the development of new learning environments is that of Learning Object. Several approaches have been proposed to generate educational material based on LOs; for instance, in [3] a system is presented for the automatic provision of web-based education.

Among the web-based tools for active learning, telelaboratories (i.e., remote laboratories accessible via Internet) are probably the most interesting in view of their pedagogical value and flexibility of use. Experiences in a telelaboratory are considered a very effective tool for a deeper learning of the techniques related to the formative objectives of an institutional course [4]. With respect to a conventional laboratory, a telelaboratory allows a more efficient management of the resources (in terms of space and time) available for students.

In this paper we describe a telelaboratory called REAL (Remote Experiments for Active Learning) Lab. The REAL Lab is a spin-off of the Robotics Lab of our department. Its objective is to allow the remote execution of robotic control experiments of educational and research interest. The rest of the paper is organized as follows. In Section 2 we briefly give an overview of existing telelaboratories and of the REAL Lab. Section 3 illustrates two typical work sessions within REAL Lab, i.e., the execution of a virtual and a physical experiment. The REAL Lab software architecture is described in some detail in Section 4, while Section 5 discusses possible future developments. Some conclusive remarks are presented in Section 6.

## **2 The REAL Lab approach**

A telelaboratory can be thought of as an environment where a set of experiments is accessible via a remote connection. Existing realizations can be divided into two categories, i.e., virtual and physical telelaboratories.

In a virtual telelaboratory, users can perform experiments on simulated systems by means of special software provided by a server. Some interesting examples are the Virtual Control Laboratory of the University of Edinburgh [5] and the Virtual Laboratory of the Department of Physics at the University of Oregon [6], which allow to simulate the behavior of several simulated dynamic systems including mechanical, thermodynamical or electric processes. Virtual laboratories provide interactive, web-based experiences aimed at increasing students' understanding of the general principles involved in the experiments. They may also be used to simulate inaccessible devices, to replace hazardous environments or to provide a visual rendering of very complex computations. A nice example of the last case are the Virtual Laboratories in Probability and Statistics of the University of Alabama [7].

A physical telelaboratory allows instead the connection with physical setups on which to perform experiments. As pointed out by Kocijancic et al. [12] and by Exel et al. [13], experimentation on physical systems is potentially more formative with respect to virtual experimentation. One of the first examples in the world was the telerobot of the University of Western Australia [8]: connected users can remotely grasp and manipulate small objects with the aid of a robotic manipulator. Other realizations, more specifically aimed at education, include the Remote Laboratory of the Stevens Institute of Technology [9], the Web Laboratories of the University of Tennessee at Chattanooga [10] and the Automatic Control Telelab of the University of Siena [11]. The latter example also illustrates how telelaboratories can be useful to stimulate a constructive competition among students: users can record the results of their experimental sessions and compare them with those obtained by others.

The REAL Lab is a virtual and a physical automatic control telelaboratory in one. Students can simulate the behavior of robotic systems under the action of different control laws and perform the same experiment on the physical systems. In this way it is possible for the control student to appreciate the quality of the mathematical model used for the simulated system as well as the effects of the various nonidealities affecting the physical system, such as unmodeled dynamics, disturbances, input/state constraints, etc.

The REAL Lab allows to execute virtual experiments on the client side by means of Java applets. With this solution, we avoid congestion of the server as well as restrictions on the number of connected users. The results of simulations are visualized as plots of the input, state and output variables and as animations of a VRML model of the system. In a physical experiment, in addition to the plots of the relevant variables, the user can observe in real-time the behavior of the system through the images acquired by a webcam. The number of physical systems which can be supported by the REAL Lab is, in principle, unlimited, thanks to its distributed software structure. The limited availability of other resources (e.g., space) may, however, generate practical constraints on the number of physical systems available for experiments. There are currently three systems available for virtual and/or real experiments in the REAL Lab.

## **3 Two illustrative sessions**

The REAL Lab can be reached at the following web address: [www.real\\_lab.uniroma1.it](http://www.real_lab.uniroma1.it). The home page (Fig. 1) allows users to access several sections. The "Presentation" section and its subsections give an overview of the REAL Lab architecture, including detailed information about the developed software and some of the adopted technical solutions, such as the realization of the vision systems associated to each experimental set-up.

From the home page it is possible to access instructions on how to use REAL Lab by selecting the "User manual" hyperlink. In particular, this section contains a manual explaining the various user interfaces.



Figure 1: REAL Lab home page

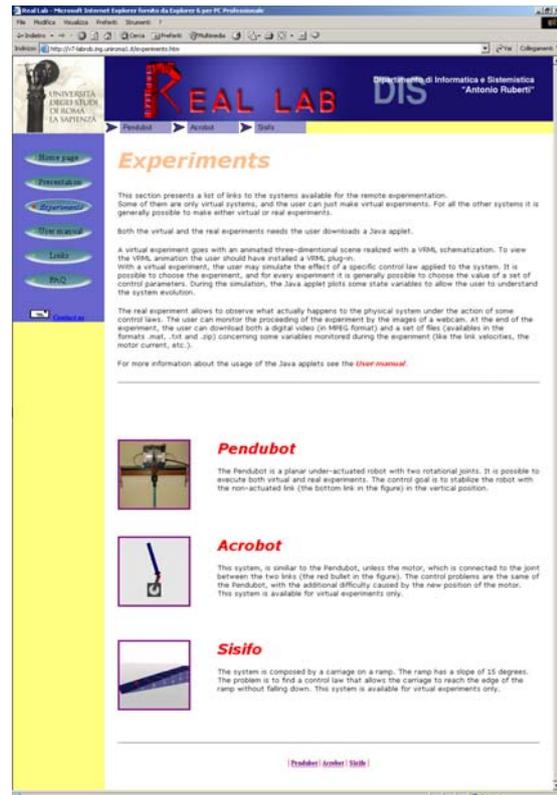


Figure 2: Experiments page

The “Experiments” page (Fig. 2) leads to the robotic systems on which to perform virtual and/or physical experiments. There are currently three mechanisms available, two of which are only virtual systems. The “Pendubot” system, instead, can be used for both virtual and physical experiments.

After choosing a system, the user reaches a web page containing links to various subsections, including a mathematical description of the system and an explanation of the control problem with the associated experiments. For example, Fig. 3 shows what is displayed if the “Pendubot” system is selected. At this point the user must choose between a virtual and a physical experiment. Clearly, if the system is not physically available (either because it does not exist or because another user is performing experiments on it), it will only be possible to work with the virtual system.

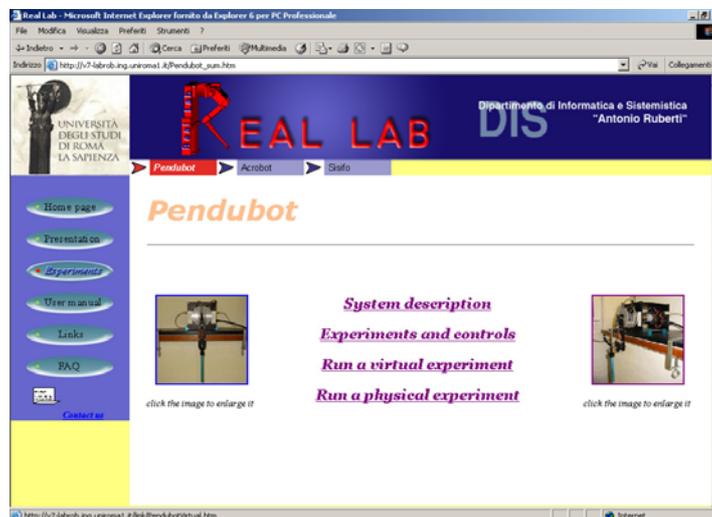


Figure 3: Pendubot home page

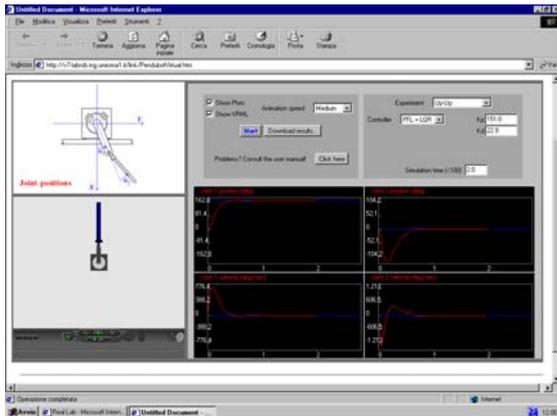


Figure 4: Java applet for virtual experiments

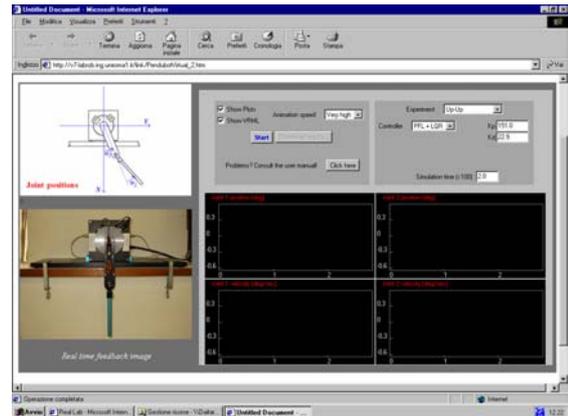


Figure 5: Java applet for physical experiments

### 3.1 A virtual experiment

Selecting the link “Run a virtual experiment” from the main page of a dynamic system (see Fig. 3 for the Pendubot system), the download of a Java applet starts automatically (Fig. 4). This applet is the client-side software provided to perform virtual experiments. Once downloaded, it does not require any connection to the server; therefore, virtual experiments do not congestion the REAL Lab. The graphical interface of the applet can be divided into four regions. In the top-left region, there is an illustrative sketch of the system, including reference frames for the relevant coordinates. In the top-right region, the user finds the main panel that controls the execution of the experiment. The user must first pick one experiment among those available. For every experiment, the applet contains a list of built-in controllers to choose from. Once a controller has been selected, the user can modify some of its parameters to understand their role in the control action.

In the bottom-left region, there is a VRML scene representing the system. This scene will be animated during the simulation to give an idea of what happens under the action of a certain control law. Finally, the bottom-right region of the applet interface contains some windows through which the user can observe the evolution of relevant input, state or output variables. The number and type of these windows may change from one system to another.

### 3.2 A physical experiment

To run a physical experiment, the user must click on “Run a physical experiment” from the main page of the selected system. As for virtual experiments, a Java applet (Fig. 5) is automatically downloaded. The interface layout is the same, with the exception that the space reserved to the VRML scene is now replaced by a real image refreshed by the server twice a second, so that the user may observe the behavior of the physical system in real-time. When the experiment is completed, the “Download results” button becomes active, making it possible to download some files containing a sampled history of relevant variables measured during the experiment. The files are also available in Matlab format. This feature may be very useful to compare the results of the physical experiment with those of a local simulation performed in Matlab/Simulink. It is also possible to download a video of the experiment in the MPEG format, to better appreciate the details of the system motion.

## 4 Architecture and technical details

The REAL Lab has the client-server architecture shown in Fig. 6 and is based on five software modules:

- The Java applets which provide the graphical interface on the client side for both virtual and physical experiments, and the simulation routine (based on Runge-Kutta integration) for virtual experiments only;
- The Web server (Apache server for Linux) which manages the web site and the remote accesses;
- The Servlets and their server<sup>1</sup> (Tomcat 4.0 for Linux) which manage the access to the physical systems and coordinate the activities of the Telelab and the Grab server;
- The Grab server which manages frame grabbing via a webcam;
- The Telelab server which directly controls the behavior of a physical system.

All the modules communicate by the TCP/IP protocol. A copy of the Telelab server must be installed on each PC controlling a physical system. The PC is connected to the Servlet server via TCP/IP. These choices allow a distributed architecture of the telelaboratory and, hence, the possibility to install various experimental set-ups in different places. In the current configuration, the Web, Servlet and Grab servers run on the same PC under Linux, while the Telelab server runs on a different PC under WinNT. Other configurations are possible with the constraint that the Linux programs (Web, Servlet and Grab servers) cannot be executed on the same PC running the Telelab server, which is a Windows program.

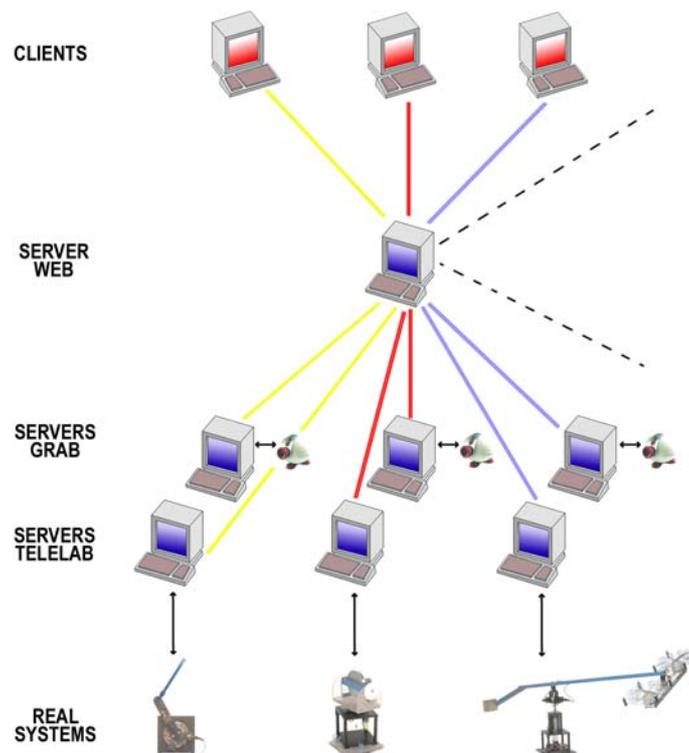


Figure 6: The REAL Lab architecture

<sup>1</sup> In Fig. 6 the Servlet server is considered part of the Web server for simplicity.

The client modules realized by the Java applets are platform-independent and do not need any specific commercial software to be installed on the client PC. The only requirement is a VRML plug-in to animate the VRML scenes during virtual experiments.

The vision apparatus associated to each physical system is composed by a webcam and the Grab server developed within the project presented in this paper. The Grab server sends the grabbed images via ftp to the Servlet, which in turn, again via ftp, sends them to the client module. The current frame rate is two images per second, already over the bandwidth limit imposed by the Internet connections; it can be however raised up to 25 frames per second. The Grab server, in fact, collects 25 images every second to realize a movie of the experiment, encoded as an MPEG file to improve the quality.

The control laws for physical experiments are implemented in Matlab/Simulink to simplify the addition of new controllers and/or new experiments. The real-time code relative to the controllers is executed by the Telelab server, which needs Real-Time Workshop and Matlab running simultaneously. The Telelab server sends to Matlab commands to change the values of control parameters, to manage and save numeric data relative to the experiments, to start or to stop the execution of the real-time code (i.e., the experiment). Communication between the Telelab server and Matlab is based on the Microsoft DDE protocol (Dynamic Data Exchange).

To run the real-time code of the experiments, the PC controlling the physical system should be equipped with a real-time operative system. An alternative solution, adopted in this work, is to use a traditional operative system with the addition of some special patches, such as VenturCom RTX, providing real-time capabilities.

Since every physical system can only be controlled by one user at a time, every experiment may take a maximum amount of time, after which the Telelab server will free the resources sending to Matlab and to the Servlet the command to stop the experiment.

The activities of the Servlet, the image grabbing of the Grab server and the management of the physical system from the Telelab server are all coordinated through a specific high level protocol based on the TCP/IP stack.

## ***5 Future developments***

Currently, the REAL Lab does not contain any mechanism for user registration or identification; therefore the access to the telelaboratory is completely free and not monitored. One of our next objectives is the realization of an identification procedure to increase security.

The REAL Lab has been designed and developed with an eye to future expansion demands. Installing new virtual experiments or new control laws is straightforward, thanks to the flexibility of the Java technology used for the applets. The installation of a physical system is clearly more time-consuming, and special attention must be paid to some hardware issues, such as the fact that the system acquisition/control board should be accessible by Simulink-compatible drivers. In the next future we are planning to install at least one new physical system.

Another important feature that will be implemented in the next version of REAL Lab is the connection between virtual and physical experiments. In particular, the user will be allowed to perform a physical experiment on a system only if the corresponding virtual experiment (with the same controller and set of control parameters) has been successfully executed in advance. This will also lead the user to appreciate the different behavior of the physical system with respect to its simulated model.

To increase the educational power of REAL Lab, we will also add some mechanisms to stimulate the competition between students, e.g., by inviting them to tune a parametric control law for a specific experiment. A public ranking could then be formed on the basis of the obtained results. The ranking could also contribute to the student's final grade.

Another feature we plan to add to the Lab is a "Forum" section, where users/students may exchange messages and comments on-line, and a repository of suggestions and questions.

## 6 Conclusions

The REAL Lab is a complex active learning tool based on the Java technology, that allows the remote execution of virtual and physical robotics control experiments. A user can either perform a virtual experiment and observe the behavior of the controlled system through a VRML animation, or perform a physical experiment and monitor in real time the controlled system through the image flow provided by a webcam. Besides, at the end of a physical experiment, the user can download a video file in MPEG format, so that he/she can review in detail the behavior of the system under a specific control law, and data files containing the sampled history of relevant variables.

The REAL Lab is based on a client-server architecture using the TCP/IP protocol; hence, the various software modules and physical systems may be installed in different servers and laboratories. The Lab has been designed so as to minimize the effort of adding new systems (virtual and/or physical), new experiments or new controllers.

## References

- [1] M.Khalifa, R.Lam,  
*Web-based learning: effect on learning process and outcome*,  
**IEEE Transactions on Education**, vol. 45, no. 4, pp. 350-356, 2002.
- [2] N.Sala,  
*The Internet and the “learning by doing” strategy in the educational processes: a case of study*,  
**1<sup>st</sup> IEEE Int. Conf. on Advanced Learning Technologies**, 2001.
- [3] F.Pantano Rokou, Y.Rokos, E.Rokou,  
*InfoLab: a web learning pedagogical based content repurposing approach*,  
**3<sup>rd</sup> IEEE Int. Conf. on Advanced Learning Technologies**, 2003.
- [4] D.A.Miele, B.Potsaid, J.T.Wen,  
*An Internet based remote laboratory for control education*,  
**2001 American Control Conference**, 2001.
- [5] University of Edinburgh,  
*Virtual Control Laboratory*,  
<http://www.chemeng.ed.ac.uk/ecosse/control/sample/map/>
- [6] Department of Physics at the University of Oregon,  
*Virtual Laboratory*,  
<http://jersey.uoregon.edu/>
- [7] University of Alabama,  
*Virtual Laboratories in Probability and Statistics*,  
<http://www.math.uah.edu/stat/>
- [8] University of Western Australia,  
*Telerobot on the web*,  
<http://telerobot.mech.uwa.edu.au/>
- [9] Stevens Institute of Technology,

***Remote Laboratory,***

[http://www.soe.stevens-tech.edu/Academics/remote\\_lab.html](http://www.soe.stevens-tech.edu/Academics/remote_lab.html)

- [10] University of Tennessee at Chattanooga,  
***Web Laboratories,***  
<http://chem.engr.utc.edu/>
- [11] University of Siena,  
***Automatic Control Telelab,***  
<http://www.dii.unisi.it/~control/act/home.php>
- [12] S. Kocijancic, C. O'Sullivan,  
***Integrating virtual and true laboratory in science and technology education,***  
**32nd ASEE/IEEE Frontiers in Education Conference, 2002.**
- [13] Exel,M., S.Gentil, F.Michau and D.Rey,  
***Simulation workshop and remote laboratory: Two web-based training approaches for control***  
**2000 American Control Conference, 2000.**