

A Remote and Virtual Lab with Experiments for Secondary Education, Engineering and Lifelong Learning Courses

<http://dx.doi.org/10.3991/ijoe.v8iS2.1984>

A. Cardoso¹, M. Vieira¹ and P. Gil^{1,2}

¹ CISUC - University of Coimbra, Coimbra, Portugal

² FCT-UNL - Universidade Nova de Lisboa, Lisboa, Portugal

Abstract—The Information and Communication Technology tools are nowadays invaluable to support e-learning and b-learning programs. The Remote and Virtual Laboratory in development at the Department of Informatics Engineering of the University of Coimbra (Portugal), RVL@DEI-UC, is a web-based platform that allows users to perform a large set of experiments in different areas and contexts, such as in education or training. This paper aims to describe the inherent potential of this platform in secondary education, engineering and lifelong learning courses. The conceptualization, architecture and implementation of the web platform for real and virtual experiments, which is remotely accessed using the Internet, are presented and the relevance of the lab's integration in an intelligent tutoring system is also highlighted, mainly in what regards the requirements of adaptation and customization to different users' profile in different learning contexts.

Index Terms—Remote and virtual lab; intelligent tutoring system; b-learning; K-12 education; engineering education; lifelong learning.

I. INTRODUCTION

The development of remote and virtual laboratories (RVLs) have received considerable attention in the last few years, specially, in terms of building web-based educational tools [1], [2], [3] and platforms for students at secondary and higher education levels, as well as for lifelong learning courses. The relevance of the integration of RVLs in intelligent tutoring systems is also of great interest, especially to incorporate the requirements of flexibility and adaptability in such systems [4].

This paradigm of learning and training enables the development of web platforms rich in educational contents and the usage, in a flexible and easy way, of remote and virtual experimental setups, breaking down spatial and temporal barriers associated with the more traditional approaches. By making use of an Internet access point different kinds of virtual and real setups, as well as, a number of idealized experiments in an e- or b-learning context may become available, which remarkably contributes to the enrichment of classroom activities [5]. Students may accomplish remotely their own experiments by just using a laptop and an available Internet connection. The reader is referred to [6] for some reflections and perspectives on the Systems and Control area of several Engineering courses.

As expected, several courses in different academic degrees are currently taking advantage of all the potential of Information Technologies, namely, the Internet infrastructure and web technologies, to design and implement distant teaching and learning paradigms. In a web-based virtual laboratory students or trainees rely on the Internet or an Ethernet local area network (LAN) to perform experiments on remote setups under the form of models. The virtual system's time response is obtained by simulation on a fast processor server [7, 8]. The outcomes of these simulations, i.e. the system time response and its inputs over time, are subsequently retrieved and analyzed by users. In the case of remote laboratories, students, instead of performing experiments on virtual plants, they carry out their experiments on remote real pilot plants, connected via data acquisition boards to a server computer [9], [10], [11]. In addition, this kind of laboratories can operate either in batch mode or online.

The integration of RVLs in an Intelligent Tutoring System (ITS) provides an effective basis to build a platform with the capability of adaptation to different users' profiles. The ITS should be able to define contents and suggest experiments according to the specific needs and backgrounds of each user. In particular, students from secondary education levels or engineering courses, and trainees from industry have different profiles and the system should be developed taking into account their specific backgrounds and requirements.

In general, the use of an ITS can be pedagogically advantageous for students and trainees, by creating adequate sequences of experiments and following up the user's performance in real time, using, for instance, task planning approaches to enhance the user's learning performance and providing feedback, help and guidance to address the user's immediate difficulties and needs [12].

The Remote and Virtual Laboratory in development at the Department of Informatics Engineering of the University of Coimbra (RVL@DEI-UC) is a laboratory with interactive computer applications that allows users to carry out remote or virtual experiments considering real physical systems or simulation models that can be remotely or locally accessed and controlled through a web browser. The experiments provided to each user through the intelligent tutoring system are selected according to its particular profile, background and the characteristics of the course in terms of contents and goals [13].

The paper is organized as follows. In Section II it is described the Remote and Virtual Lab of the Department of Informatics Engineering, focusing on its main components and functionalities. Section III is devoted to the Intelligent Tutorial System describing some features and requisites of this platform, while in Section IV sketch some experiments that can be considered in secondary education, engineering and lifelong learning courses. Finally, in Section V some conclusions are drawn.

II. THE REMOTE AND VIRTUAL LAB

The Remote and Virtual Lab (RVL@DEI-UC) is a hybrid laboratory with interactive computer applications that allow users to carry out real or virtual experiments on systems that can be remotely or locally controlled through a web browser using shared communication networks.

A. The RVL Conceptualization

The RVL@DEI-UC is a web based platform that enables carrying out remote and virtual experiments in a real laboratory, using shared data networks. As an example, every user has the possibility to configure and follow the system's behavior, as well as the performance of local controllers, like in most systems of this nature. However, this approach offers to users, considerably more powerful tools, since users can also design and run their own controllers. This concept is quite different from classical approaches and falls under the category of a relatively new hybrid scheme where the controller's parameters are tuned on-line on the client side, while the server delivers the system's response.

The advantages of this design are, among others, the ability for a client to use the platform to test controllers developed by him/herself, giving additional usability to the system as a resource for investigators in different areas, such as, remote system identification, networked control systems, distributed systems or fault tolerant control, just to name a few; the ability to use the computational power of clients' computers to implement more resource demanding control laws; the ability to use complex controllers despite significant limitations on the server side; the possibility of privacy for remote users who may not wish to send their controllers code for compilation and execution to the server; more flexibility in the controller design process, since the user has the freedom to implement any controller on the client side; better security against download viruses or malware at the server side, since the server does not compile or run any remote software locally.

However, this paradigm poses a number of completely new challenges concerning communication failures and latencies. For this reason the platform was designed to include fault-tolerance mechanisms, which can address the problems derived from the unreliability of the Internet and the extra flexibility and responsibility that is presented to users. Additionally, the platform offers a wide range of monitoring features, such as: numeric and graphical visualization of information related to running experiments and the network communication quality of service; animated representation of the experiment, which can be shown in 3D, 2D or both; video and audio streaming of the experiment; the possibility to interact with a local operator or system's manager, if available, through a microphone.

As mentioned, some of these features are not common in systems of this kind, namely the ability to access predefined performance metrics to analyze the quality of service of the network communication. Distinctly from most platforms, any user has the chance to use available monitoring features interchangeably or having them combined, for example, placing the animation on top of the video image in order to compare de accuracy of the virtual representation.

B. The RVL Architecture

The RVL@DEI-UC is based on a client-server architecture (see Figure 1) and consists of four main applications. The server application runs on a server computer at the laboratory site and each client application is installed on a remote computer. Furthermore, the media server application is located in an additional computer, while the process application runs on each computer connected to a real process via an I/O board.

The server application establishes the interaction between either the real setup or the virtual process and the client application, which is located on a remote computer. This application is implemented in JAVA and makes use of MATLAB functions, namely, for I/O communication with DAQ cards or to evaluate the sequence of control actions, taking into account previous inputs and outputs, as well as reference trajectories. In case the platform is configured as a Virtual Laboratory the computed sequence of control actions is provided to the virtual plant environment, and subsequently used in the plant's simulation model by calling the MATLAB computational engine.

Each real process is connected to an off-the-shelf PC equipped with input/output (I/O) interface boards from different manufacturers. The software used to interact with laboratory systems is based on Matlab/Simulink, including the Humusoft Extended Real-Time Toolbox and considers safety measures to guarantee the setup integrity against misuses.

The client application is launched by a user logged on in a remote computer, connected to the server via an Ethernet local area network or the Internet. After the authentication, the client application gets from the server a list of available real and virtual plants. For example, in a control experiment, the remote user should select and parameterize a given controller to run on the server, in case of batch operation mode or, alternatively, on the remote computer, in case of online control scheme. In this topology the computed sequence of control actions is sent to the server using a TCP connection. Users should also define other parameters, such as, the sampling time, the overall time for the experiment, the reference trajectories and the type of control cycle.

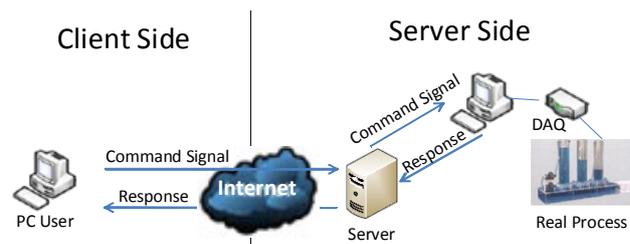


Figure 1. Main client-server interactions in an experiment to remotely control a real process.

While the experiment is in progress, the remote user can follow the closed loop control performance in a dedicated window, displaying outputs, reference trajectories and computed control actions. Additionally, the user may have access to a number of monitoring statistics with respect to the running experiment. These statistics include, updated number of timeouts, server-client and client-server latencies, just to name out a few.

C. The web based platform

To use the RVL, the user should log on in the platform's website and go through the requirements verification page to investigate whether he/she has the s required software. Figure 2 shows the main webpage.

After choosing the type of process, real or virtual, the website lists the available processes on the database and presents those that are ready to be used. For selecting and start running an experiment that is available (Figure 3), it is previously necessary to set the experiment's configuration parameters.

In the next step, the user called to configure the experiment (see Figure 4), which includes, for example, the control horizon, the sampling period and the contingency plan, in case of a problem with the communication network [14]. Next, users have to define and configure the process controller by selecting and initializing a given controller through the controller configuration window (see Figure 5). Only after the previous steps are completed users are allowed to start running the experiment.

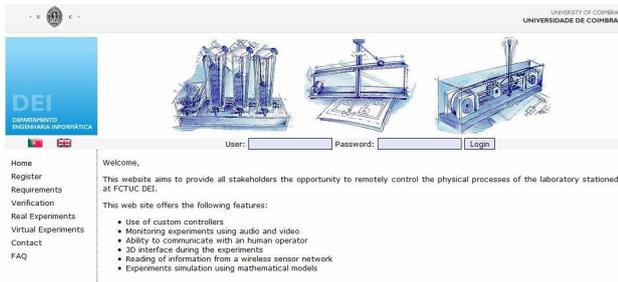


Figure 2. Main web page of the platform.



Figure 3. Example of the experiment's selection interface.



Figure 4. Connection configuration window.

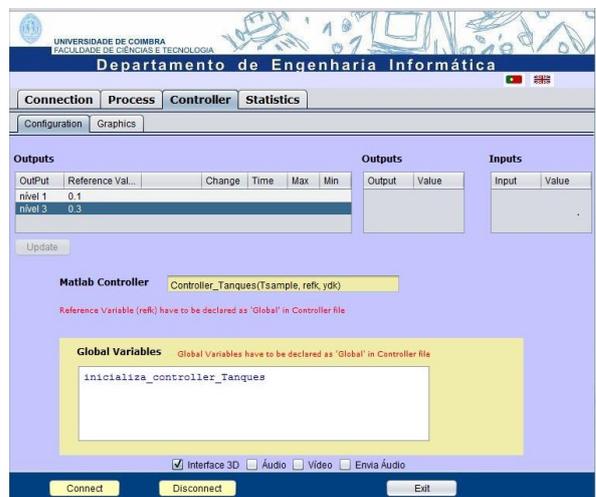


Figure 5. Controller configuration window.

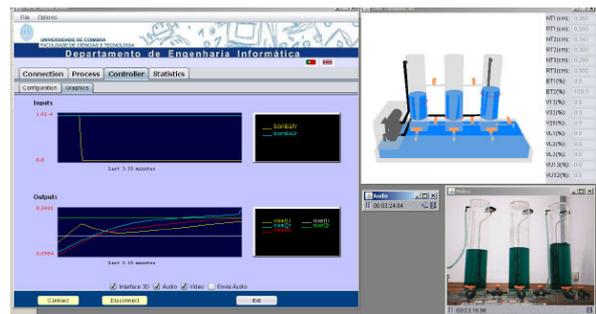


Figure 6. Platform interface for visualization of an experiment concerning the control of a three-tank system.

During experiments users can follow the closed loop control performance in a numeric window, as well as in a graphical window that displays the outputs, reference trajectories and the computed control actions. Additionally, they may have access to a number of monitoring statistics regarding the running time, along with, some monitoring features: 2D/3D model and video and audio streaming. Figure 6 shows an example of the described monitoring features in the context of a three-tank system experiment, where a real-time graphic representation of the system's inputs and outputs audio, video and 3D dynamic visualization is presented.

The platform's interface from the user's point of view includes the following means of visualization:

- Numeric visualization of signals associated with the process and network communication control metrics;
- Graphical visualization of system's inputs and outputs and network communication control metrics;
- Dynamic virtual representation of the experiment and associated input and output variables' values;
- Video and audio streaming.

The real-time numeric visualization of signals related with the process inputs and outputs is complemented with the graphical visualization of these variables, which allows the observation and analysis of the entire experiment progress. The numeric and graphical visualization of network communication performance metrics (Figure 7) is not commonly available at the client side in this kind of platforms. Nevertheless, it is an important feature that allows users to evaluate the network communication's performance, validating the quality of service, detecting possible bottlenecks along the control cycle loop (forward and feedback) and to take advantage of acquired data to fine-tune his/her controller to adapt to changes in the network conditions.

The virtual representation of the system and its related variables and parameters is quite significant in virtual experiments, where video streaming is not inherently available, simulating dynamically the evolution of the system's variables and events taking place in a 3D environment. Nonetheless, this representation can also be accessed in remote experiments interacting with real setups and can be combined with the live video feed. Additionally, it is possible to overlap these two forms of visualization so the user can better assess the accuracy of the virtual representation.

The audio and live video feeds are invaluable for making the remote experience as close to an actual on-site experiment as possible. These elements along with the remote control over the experiment and the ability to extract data results are the main components, and the most important tools offered by this remote and virtual laboratory.

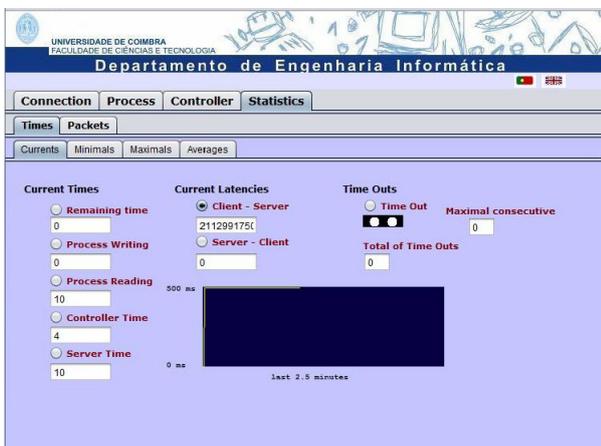


Figure 7. Platform interface for visualization of network communication performance metrics.

III. THE INTELLIGENT TUTORING SYSTEM

The interaction with laboratory plants could be defined in function of a particular user's profile or preconfigured according to a scheduled learning process. For instance, depending on the user's skills, the control system can be built by using a remote controller designed and implemented by the user on the client side, or considering a local controller with a predefined structure, but in which some of its parameters can be user configured remotely. The set of experiments that can be performed through the RVL includes various processes with different levels of complexity that are chosen in an adaptive way, in order to fulfill the objectives and the characteristics of each learning program.

The integration of remote and virtual laboratories in intelligent tutoring systems is of great relevance for the learning process because users can have the opportunity to complement the theory with practice, which is commonly based on only simulators. As such, they are actively involved in developing and implementing control systems to interact with real or hybrid laboratory processes.

Moreover, building courses with requirements of adaptability to different profiles of students/trainees (K-12 students, engineering students or trainees in lifelong learning) can improve greatly the learning process, by motivating users and giving them conditions to knowledge acquisition with a priori guaranteed good results.

The tutoring system under development at DEI-UC seeks to incorporate intelligent agents that implement mechanisms of identification, monitoring and supervision of users' behavior and progress, namely, in terms of performance and accomplishment of tasks proposed in each course's module.

The ITS is supported on a Moodle platform and is being developed using the freeware CourseLab authoring tool. Figures 8 and 9 present examples of slides where users can obtain information about a particular experiment and interact with the remote and virtual lab, such as running an experiment, for instance to control a three-tank system in the context of an engineering course.

Whenever the ITS identifies the necessity to popup suggestions, warnings or guidelines to a user, an intelligent agent shows a virtual teacher interacting with that user in a number of ways, such as, by writing a message on a slide.

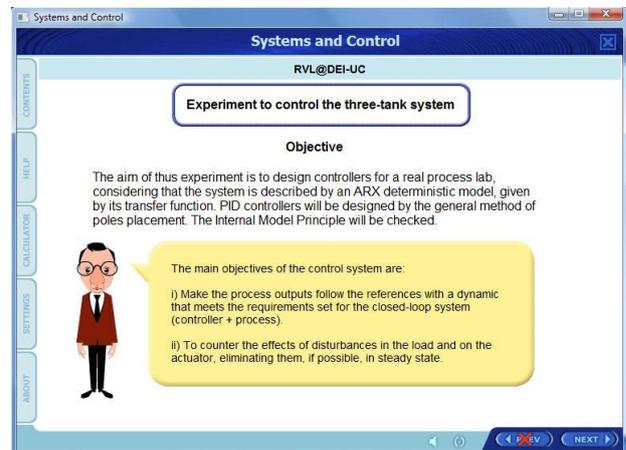


Figure 8. Example of the introduction to an Experiment.

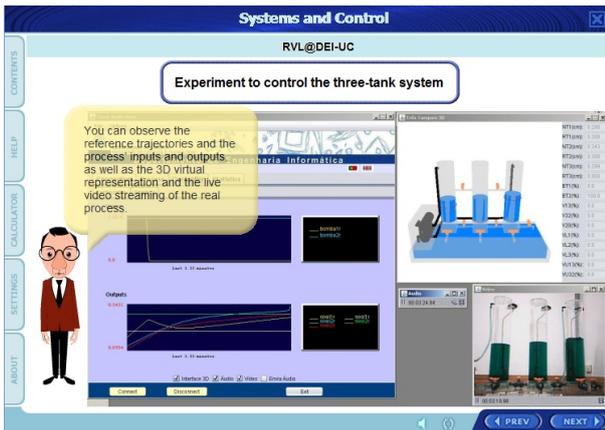


Figure 9. Systems and Control course interacting with the RVL in a three-tank system control experiment.

The methodology, behavior and intelligence of agents are still in process of investigation, having in mind that the quality of the system depends deeply on the agents' ability to interpret the behavior and progress of each user and to react accordingly. On the other hand each intelligent agent is also intended to be evaluated in terms of improvements achieved by users in the course of their learning activities, and using assessing metrics. Based on these scores agents are also trained in a way to improve their own tutoring skills.

IV. EXPERIMENTS

Several real-time and virtual experiments can be carried out using the RVL@DEI-UC platform, enabling the interaction with different laboratory systems, while complying with common control bandwidth available in shared communication networks. Usually, the laboratory experiments are firstly analyzed and simulated on a Matlab/Simulink environment comprising a number of toolboxes, namely, the Signal Processing, the Control System, the System Identification, the Image Processing and the Symbolic Math Toolboxes. Next, the user has the possibility of run experiments on-site or remotely using the platform.

Some of the experiments for K-12 education, engineering and lifelong learning courses are briefly described as follows.

A. Experiment 1: Hydrostatic and hydrodynamic

The purpose of this experiment is to analyze and observe different aspects and laws of hydrostatic and hydrodynamic, at a secondary education course level, such as Physics. The two-tank system (model PCT9 from Armfield) is one of the systems considered in this experiment. This system (Figure 10) relies on a motorized valve to control the flow rate of water pumped from a sump tank to a dual compartment tank at the top of the plant, where a sensor measures the liquid level. Several outputs with different flow rates can be chosen.

This process could be used to demonstrate the laws of hydrostatic and of hydrodynamic. In hydrostatic, the students can realize the notion of fluid, volumic mass or its fundamental law. In hydrodynamic, some experiments can be conducted to observe the movement of fluids, the laws of conservation of mass, the Bernoulli's equation or other laws of hydrodynamic.

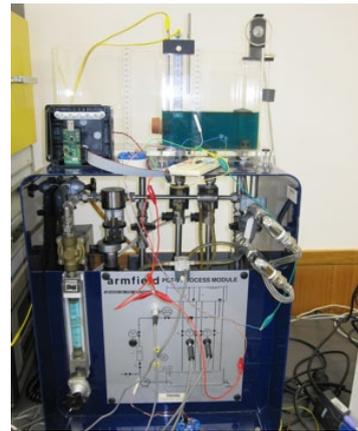


Figure 10. The two-tank process (PCT9) used in Experiment 1.

B. Experiment 2: Networked Control Systems

This experiment aims to incorporate a non-dedicated communication network between the controller and the system to build a networked environment, in the context of an engineering course level. Using the Internet to remotely control a real system, it is possible to analyze the robustness of the overall control system to delays and faults in the communication network.

The three-tank system (Amira - DTS 200) is a benchmark system that is used to implement a networked control system. This laboratory process (Figure 11) is a nonlinear multi input-multi output (MIMO) system with three level sensors and two actuators (pumps).

In this experiment setup a user can implement and test a remote controller (on the platform's client side) that receives the tanks' levels and sends the control actions to the pumps through an Internet connection and local server. Additionally, he/she has the possibility to implement, for example, a predictive controller that generates and sends current and future control actions to support contingency plans for situations where the communication network fails.

C. Experiment 3: Distributed Control Systems

In this experiment a Supervision and Distributed Control System (DCS) over a Wireless Sensor and Actuator Network (WSAN) is considered in the context of lifelong learning courses. Several aspects concerning supervision systems, wireless sensor and networks communication protocols are studied and a distributed control system is implemented and tested.

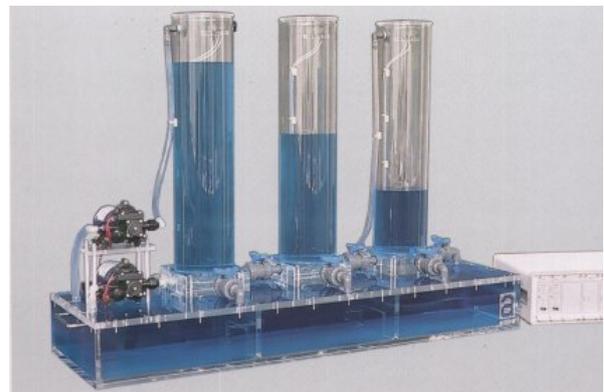


Figure 11. The three-tank Process (DTS200) used in Experiment 2.

