

# RemoteLabs Platform

<http://dx.doi.org/10.3991/ijoe.v8iS2.1900>

N. Crabeel<sup>1,2</sup>, B. Campos Neves<sup>2</sup> and B. Malheiro<sup>2</sup>

<sup>1</sup> Katholieke Hogeschool Sint-Lieven (KAHOSL), Ghent, Belgium

<sup>2</sup> Instituto Superior de Engenharia (ISEP), Porto, Portugal

**Abstract**—This paper reports on a first step towards the implementation of a framework for remote experimentation of electric machines – the RemoteLabs platform. This project was focused on the development of two main modules: the user Web-based and the electric machines interfaces. The Web application provides the user with a front-end and interacts with the back-end – the user and experiment persistent data. The electric machines interface is implemented as a distributed client server application where the clients, launched by the Web application, interact with the server modules located in platforms physically connected the electric machines drives. Users can register and authenticate, schedule, specify and run experiments and obtain results in the form of CSV, XML and PDF files. These functionalities were successfully tested with real data, but still without including the electric machines. This inclusion is part of another project scheduled to start soon.

**Index Terms**—Remote experimentation; Web application; client server architecture; electric motor; electrical engineering education.

## I. INTRODUCTION

Electric machines are an important subject of the Electrical Engineering degrees. The students of the BSc. and MSc. Electrical Engineering – Power Systems as well as the BSc. in Electrical Engineering and Computers are required to undertake several electric machines related modules and conduct a high number of experiments at the Electric Machines Laboratory, leading to an intensive use of this laboratory.

This project is a first step towards the development of a remote experimentation platform intended to allow students, teachers and technicians to perform experiments with electric machines anytime and anywhere. The goal is to design and develop a Web application for the remote experimentation of an electric motor.

## II. BACKGROUND

### A. Remote Laboratories

Remote laboratories have been introduced during the last decades into engineering education processes as well as integrated within e-learning frameworks offered to engineering and science students. Remote laboratories are also being used to support lifelong learning and student's autonomous learning activities. Standardisation efforts in this domain are being conducted by the Global Online Laboratory Consortium (GOLC) [1], a non-profit organisation dedicated to the definition of a unified interoperable architecture and a common infrastructure for remotely accessible laboratories. lab2go [1] maintains a repository of existing online laboratory platforms, such as iLabs [3],

Weblab-Deusto [4], OCELOT [5] or Labshare [6], and a list of available online experiments.

In particular, we are interested in remote laboratories used for teaching power electronics or electric machines controlled through drives. Education in power electronics requires a background in a wide range of disciplines, such as physics, electric circuits, analog and digital electronics, as well as informatics and automatic control. Valuable training experiments involve the addition of loads to the power electronics circuits such as different electric machinery. Hence, most power electronics laboratories, remote or on-site, use electric drives. These subjects have been gaining increased attraction among researchers nowadays due to escalating need for alternative energy sources and related generation and distribution efforts. According to [7], some of the most representative remote laboratory systems in this domain are:

- Remote Internet-based laboratory for experimentation with a 120 kVA multilevel power converter: The key feature of this laboratory is the utilization of a multilevel converter that can be controlled and supervised remotely in a secured way. A wide variety of laboratory experiments are supported, from grid connection to AC motor control. The users can choose the control structure, the control parameters and the kind of load and get the graphical results of the measurements, all in real time [8].
- Web-based teaching of electric drives using a mechanical load simulator: The developed system allows testing the behaviour of an electric machine for different load types. Several typical loads in which the torque depends on time or speed are implemented. Using this tool it is possible to study the dynamic behaviour of an electric machine. It is also used to study induction machine fault diagnosis under different load types. The remote laboratory is based on an I/O interface controller and client-server architecture. This system was developed in MATLAB [9].
- Remote Laboratory Platform for Electrical Drive Control Using Programmable Logic Controllers: This platform is intended to be used in industrial automation practical work. It is accessible through the Internet, includes programmable logic controllers that can be programmed with the open software CoDeSys and includes various sensors and actuators used in industrial installations [10].

There are many other remote laboratories efforts, but the ones listed above are the most recent ones in our application domain. In this field, remote laboratories are enabling intensive use of university facilities and aiding professors and students.

### III. DEVELOPMENT ENVIRONMENT

Web development is a dynamic field populated with many server-side and client-side technologies, including languages, Web servers and database servers. As a result, the first task that confronts a developer is to sensibly choose the set of servers, languages and plug-ins that best fit the requirements and the production environment of the future Web application.

The choice of the development language is a critical step to accomplish this project successfully. For the server-side language, we chose PHP [11] because it is an open source, platform independent free solution [12]. For the client-side programming, we chose the standard Web technologies: HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) [13] for end-user presentation and JavaScript and Asynchronous JavaScript and XML (AJAX) [14] for end-user dynamic interaction. CSS is used to define the presentation style of the HTML documents, JavaScript for form checking and AJAX for event-driven asynchronous updating.

The selected Web server was Apache HyperText Transfer Protocol (HTTP) server [15] since it is an open source, platform independent Web server supporting PHP.

For structured data storage and management we adopted the open source MySQL database server [16] and the phpMyAdmin [17], a free software tool that provides a graphical user interface to interact with MySQL. Additionally, experiment data is also stored in Comma-Separated Values (CSV) and eXtensible Markup Language (XML) text files. The XML text files are compliant with a specified XML Schema Definition (XSD) [18].

For the client-server application that provides the interface with the electric machines drive, we used C#.

To install the selected Web development environment, we used WAMP [19], a free package composed of open source programs. WAMP is an acronym formed from the initials of the operating system – Microsoft Windows – and the main components of the package: Apache HTTP Web server, MySQL database server, including phpMyAdmin, and PHP Web development language.

Some additional packages and plug-ins were also installed: the PHP Captcha Security Images [20], the PHP Simple Mail Transfer Protocol (SMTP) class [21], the PHP Portable Document Format (PDF) Library [22], jQuery [23], Google Maps Application Programming Interface (API) [24] and the PHP Plonk Library [25].

Finally, we used the following Integrated Development Environments (IDE): NetBeans [26] for the development in PHP of the Web application and Microsoft Visual Studio [27] for the development in C# of the client-server motor interface application. TinyMCE [28], an open source platform independent Web-based JavaScript HTML What You See Is What You Get (WYSIWYG) editor, was used to create HTML code and to style and control user input fields.

### IV. REMOTELABS PLATFORM

The backbone of the RemoteLabs platform consists of a Web application, a motor interface application and a database. The motor interface application is a client-server application that is replicated for each motor added to the RemoteLabs platform. In particular, a dedicated server module is installed in a platform physically connected to each motor drive, resulting in a modular, scalable and distributed approach.

#### A. Architecture

Figure 1 presents an overview of the RemoteLabs platform architecture. To interact with the platform, the user needs a fixed or mobile device, a Web browser and Internet access.

All components play a well-defined role:

- The Apache application server hosts the Web application that provides the user interface. Users can register, authenticate and, upon successful authentication, schedule, create and run experiments and, finally, collect the results.
- The user and experiment data are stored in a central database and in XML files stored centrally and locally.

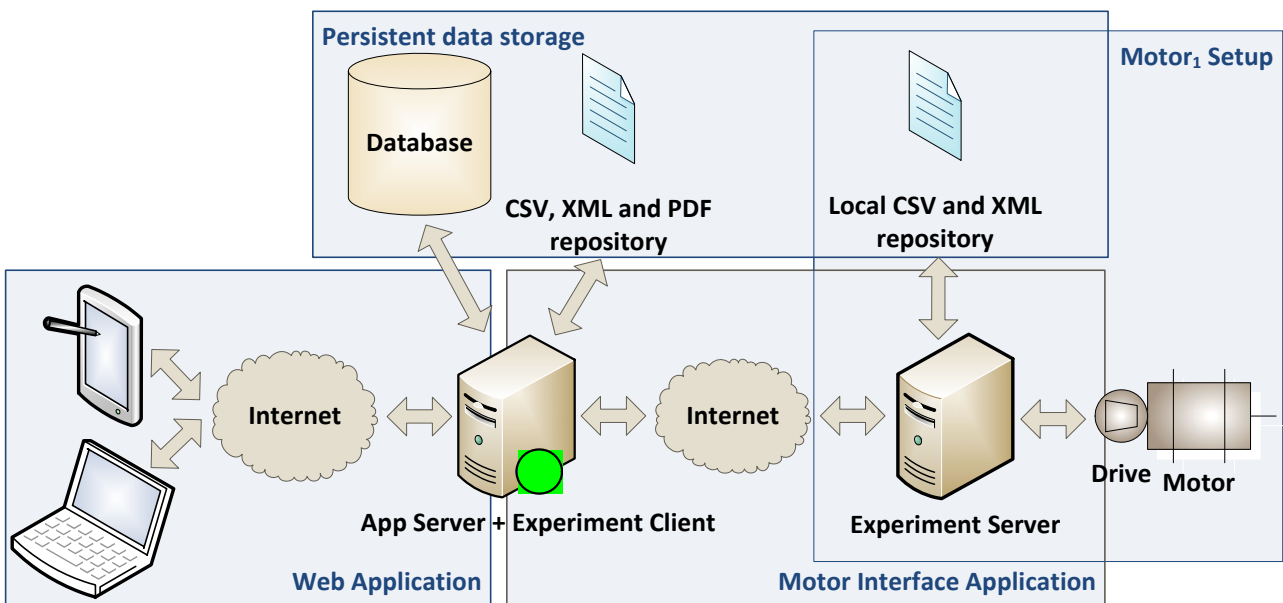


Figure 1 RemoteLabs platform architecture

- The Web application, upon the creation of an experiment on an available time slot, launches the motor interface client module.
- The client module submits the experiment input parameters to the motor interface server module.
- The server module processes the experiment inputs and interacts with the motor drive. The motor runs the experiment.
- Once the experiment finishes, the motor drive reports the results to the server module that, in turn, sends back the results to the client module.
- The client module sends the results back to the Web application, which stores them in the database and generates the corresponding XML files.
- The user obtains the result page with links to the PDF and CSV documents holding the experiment results, which include the input/output parameters, pictures and graphs.

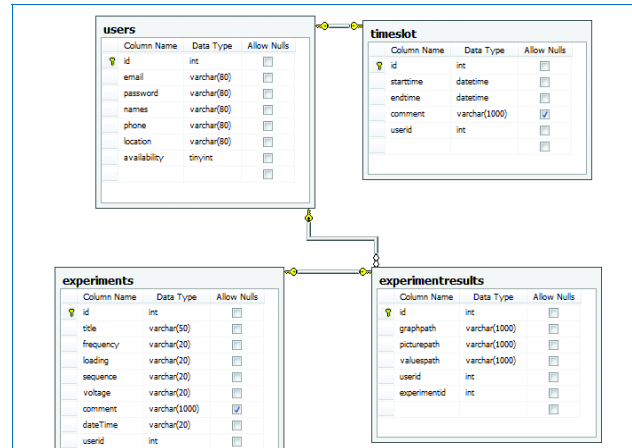


Figure 2 Database Structure

### 1) Persistent Data Storage

The created database, called remotelabs, is presented in Figure 2.

The remotelabs database structure is composed of four tables:

- The users table to hold the general user data;
- The experiments table to store the experiment configuration data, namely, the input parameters;
- The experimentresults table to hold the experiment results data;
- The timeslot table to store the scheduled experiment reservations.

As safety measure, there is database authentication and the user passwords are created with an md5 cryptographic hash.

Additionally, for each experiment conducted, an XML file is created to hold the experiment input parameters and output results. All experiment XML files are compliant with the XSD presented in Figure 3.

### 2) Web Application

The development of the Web application involved the creation of: (i) the underlying HTML and CSS code; (ii) the remotelabs database using phpMyAdmin; (iii) the PHP templates containing the HTML code; (iv) the static and dynamic pages and user session management; (v) PHP database interaction functions; (vi) scripts and a navigation flow without dead-ends; (vii) additional functions to read/write text files (XML and CSV), to generate dynamic PDF documents, launch external applications and include static Google Maps to show, by default, the Geo IP location of the user or, alternatively, the user-defined location; (viii) refined user input controls such as a date picker for experiment scheduling or styled text areas.

### 3) Machine Interface Application

The interface between Web application and machine is implemented through a client-server distributed application. The main purpose of this module is to establish a TCP connection over the Internet between Web application and machine drive to send/receive data in a safe and multithreaded way.

The client and server modules are intended to run in background on distinct platforms: the client on the Web

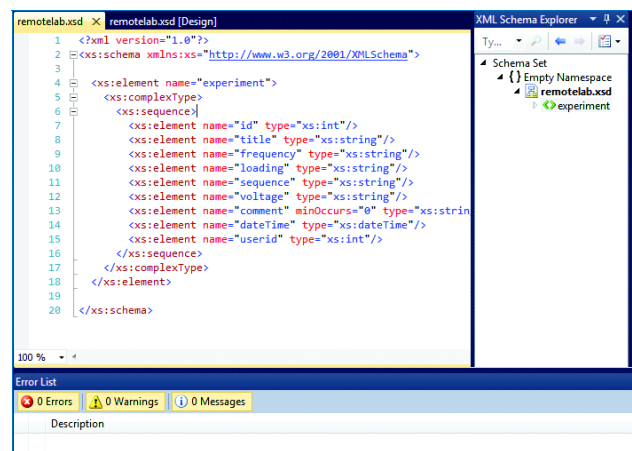


Figure 3 Experiment Data XSD

application platform and the server on the platform that physically interfaces with the electric machine.

The server module is composed of two classes:

- Server that starts a client listener and, for each new client, creates a new ClientThread instance.
- ClientThread that establishes a full-duplex TCP communication streams with the client module, processes and answers the incoming client messages and closes the communication streams at the client request.

The client module includes three classes:

- The Client class that establishes the connection with the server module, creates a full duplex communication and submits the input experiment data.
- The MessageThread class that receives and processes the answers from the server module, including the output experiment data.
- The Motor class that receives and writes the output experiment data received to a text file on the local client machine.

### B. Functionalities

The Web application provides the following functionalities:

- General: register, authenticate, view and edit the user profile, create and delete the user account and schedule, create, view and delete an experiment.

- Internal: launch external applications, database interaction, code behind templates, session management, form checking, read and write text files (XML and CSV).
- Additional: generate on-the-fly PDF experiment files and include static Google Maps with the user location.

The motor interface application implements the following functionalities:

- Client-side: connect and disconnect to and from the server, send and receive experiment data and read and write from and to text files.
- Server-side: listen, accept, multithread, disconnect clients and receive, process and answer experiment requests.

## V. TESTS AND RESULTS

Although the RemoteLabs platform is fully functional, it still does not interact with any electric machine due to unexpected problems with the release of the drive software API and licenses. For this reason, to prove our concept, we defined a set of motor experiments, run these experiments and collected the results. We then used these data to simulate actual remote motor experiments based on pre-defined input configurations, pictures and graphics.

### A. Tests

In this scenario, we prepared three possible start-up motor tests at a 50 Hz frequency: (i) star-load configuration at an input rated voltage of 230 V; (ii) delta-load configuration at an input rated voltage of 230 V; (iii) star/delta-load configuration at an input rated voltage of 230 V. The motor and its features are presented in Figure 4.

Since these three tests are very similar, we will only describe one in detail: a delta-load start-up experiment.

Suppose that a new user wishes to schedule the motor to run a delta-load experiment and print the experiment results to hand in to his/her teacher. These are the flow execution steps:

1. The new user registers at the RemoteLabs initial page. After registration, a confirmation email message is sent to the user email address. The registration form includes form checking and is protected with a Captcha.
2. The registered user authenticates and, upon successful authentication, is redirected to the RemoteLabs user home page, where the latest news, the about information and the experiment links are displayed – see Figure 5.
3. The authenticated user follows the “Reserve the Motor” link on the home page and is redirected to the scheduling page, where he can select a two hour slot in a given day. If the desired time slot is available, the user is then redirected back to the RemoteLabs home page. Otherwise, the user is invited to select another time slot.
4. During the user attributed time slot, a new experiment button appears on the RemoteLabs home page of the user. After pressing this button, the user is redirected to the add experiment page where he can select any of the pre defined experiment configurations – see Figure 6. Once the new experiment configuration is chosen, the Web application launches automatically in background the motor interface client module.

**Induction Motor Squirrel Cage MV 1009-405**  
 A 3-phase squirrel cage motor is mounted on a 10 mm thick anodized aluminium plate to be placed on the machine bed MV 1004.

General Data	50 Hz	60 Hz
4 pole machine	1.1 kW 1400 rpm	1.1 kW 1700 rpm
Star (Y)	380-415 V, 3.0 A	380-415 V, 3.0 A
Delta (D)	220-240 V, 5.2 A	220-240 V, 5.2 A
Dimensions	355 x 300 x 310 mm	Shaft height 162 mm
Weight	19 kg	

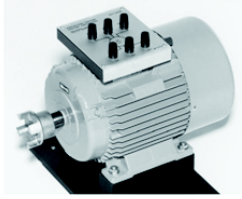


Figure 4 Motor Features

Figure 5 RemoteLabs user home page

Figure 6 RemoteLabs add experiment page

5. The client sends the new experiment configuration data to the server module and waits for the answer.
6. The server sends back the static result data corresponding to the selected experiment configuration.
7. After receiving the experiment results, the client disconnects automatically from the server and reports the experiment results back to the Web application.
8. As soon as the Web application receives the experiment results, the user is redirected to the RemoteLabs home page. Under the “View and Start your experiments” link the user finds his updated personal list of experiments, containing the newly created experiment link (Figure 7). To access any previously performed experiment, the user clicks on the experiment link and is redirect to the selected experiment details page.
9. The experiment details page (Figure 8) provides all the information available on the experiment, including user information and the experiment configuration and result data. This information is download-

able either as a CSV (Figure 9) or as a PDF (Figure 10) file. In the case of the PDF file it also includes complementary graphics and pictures of the results and the motor display, respectively.

**B. Results**

We presented the tests available and, in detail, the execution flow and results obtained when choosing a specific motor experiment – a delta-load experiment.

Only the most relevant functionalities of the Web application were explained, omitting less relevant pages that were implemented. We envisage the necessity of performing some adjustments when the actual electric motor is connected to the RemoteLabs platform.

The scalability of the RemoteLabs platform is supported by the adopted distributed and modular architecture: (i) the inclusion of a new electric machine is achieved through the installation of dedicated server module at the machine premises; and (ii) the number of experiments provided depends solely on the functionalities offered by the drive protocol.

**VI. CONCLUSIONS**

The developed RemoteLabs platform proves the feasibility of the concept and provides the grounds for further developments. In fact, this project is the first building block of a wider, more ambitious goal – the creation of a remote Electric Machines Laboratory infrastructure. The remote laboratory approach allows intensive, controlled, safer and easy use of expensive equipments over the Internet.

**A. Achievements**

The RemoteLabs platform provides the basic functionalities to set up a remote motor experiment by implementing all the functionalities originally specified and beyond.

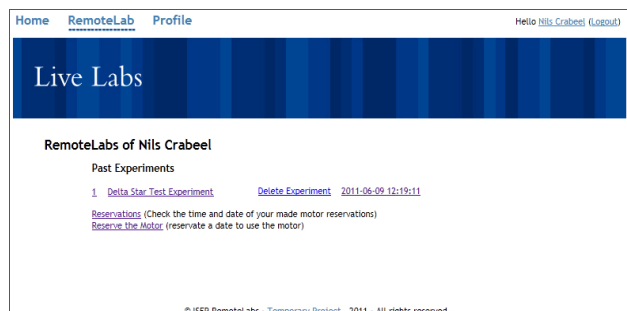


Figure 7 RemoteLabs experiment page

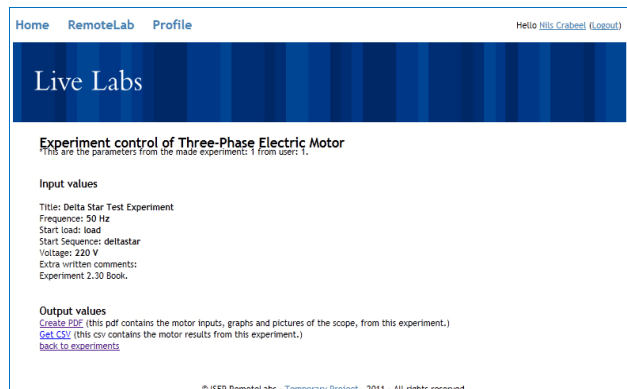


Figure 8 RemoteLabs experiment details page

This platform is composed of a front-end Web application where authenticated users can reserve time slots and specify typified experiments for a drive controlled motor. Once a new experiment is submitted, the motor interface application is invoked and the experiment parameters are sent to the remote service that physically interfaces with the motor drive. The results of each experiment are sent back over the reverse path and are stored in a database and in XML files. Finally, the Web application presents users with the results in the form of dynamically created PDF or CSV documents.

At this moment, the project is testable only with stored real motor data. As soon as the platform of the motor interface server module is physically connected to the motor drive and the communication protocol between server and drive implemented, experiments can be remotely performed on the motor.

This approach can be used by students and professionals alike to improve their knowledge and understanding of electric motors.

```

"Title"      ":" "Current";
"ID"        ":" 1;
"Type"      ":" "Envelope";
"Date"      ":" 01/21/95;
"Time"      ":" 23:48:55;
"X Scale"   ":" 5,0E+00;
"X At 0%"   ":" -1,00E+01;
"X Resolution" ":" 2,500000E+01;
"X Size"    ":" 254;
"X Unit"    ":" "s";
"X Label"   ":" "5 s/Div";
"Y Scale"   ":" 1,0000E+01;
"Y At 50%"  ":" 0,E-03;
"Y Resolution" ":" 2,506266E+01;
"Y Size"    ":" 256;
"Y Unit"    ":" "A";
"Y Label"   ":" "A";

-1,00E+01;-5,27E-01;6,70E-01
-9,8E+00;-5,27E-01;6,70E-01
-9,6E+00;-5,27E-01;6,70E-01
-9,4E+00;-5,27E-01;6,70E-01
-9,2E+00;-5,27E-01;6,70E-01
-9,0E+00;-5,27E-01;6,70E-01
-8,8E+00;-5,27E-01;6,70E-01
-8,6E+00;-5,27E-01;6,70E-01
-8,4E+00;-5,27E-01;6,70E-01
-8,2E+00;-5,27E-01;6,70E-01
-8,0E+00;-5,27E-01;6,70E-01
-7,8E+00;-5,27E-01;6,70E-01
    
```

Figure 9 CSV experiment file

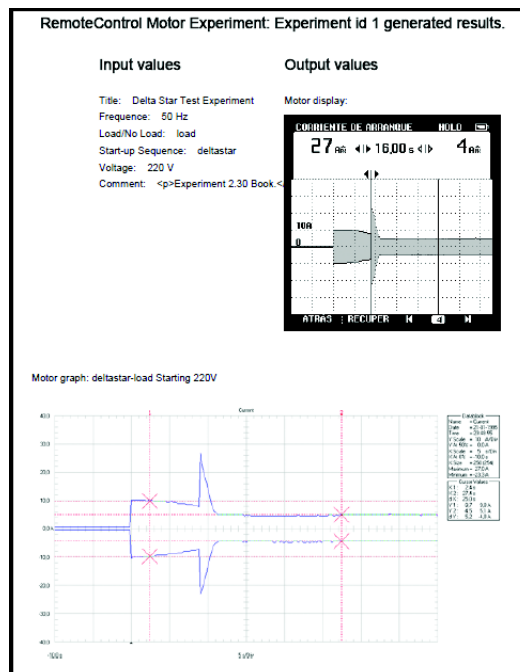


Figure 10 PDF experiment file

### B. Future Developments

This project, as the basis of a larger framework, has promising perspectives. The first suggestion is, naturally, to interface the platform with a real motor so that actual remote experiments can be conducted. At this stage, we envisage the necessity of performing some adjustments to the RemoteLabs platform in order to accommodate possible real world hard constraints. Additional devices can be added to improve the remote motor experiment experience, such as a web camera and a microphone, and to report additional information such as complementary measuring devices. Figure 11 illustrates such a scenario.

Finally, the major plan is to add to the platform not just three phase induction electric motors, but all available electric machines equipped with electronic drive controllers. This requires a considerable time and effort in drive interface protocols implementation and experiment typifying.

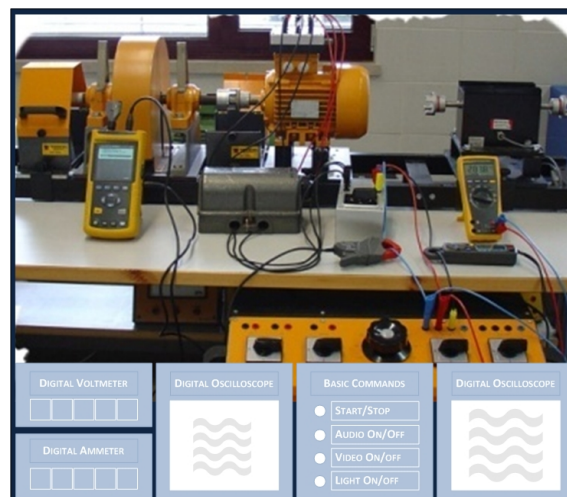


Figure 11 RemoteLabs future scenario

### ACKNOWLEDGMENT

The authors would like to thank their home institutions, ISEP and KAHOSL, and to the EU Lifelong Learning Programme, in particular, the Erasmus student mobility programme.

### REFERENCES

- [1] Global Online Laboratory Consortium (GOLC). Accessed in September 2011. Available at <http://www.online-lab.org/>.
- [2] lab2go, Center of Competence in Online Laboratories and Open Learning, Carinthia University of Applied Sciences, Villach, Austria. Accessed in September 2011. Available at <http://www.lab2go.net/>.
- [3] iLabs: Internet access to real labs, Massachusetts Institute of Technology, MA, USA. Accessed in September 2011. Available at <http://icampus.mit.edu/iLabs/>.
- [4] WebLab-Deusto, University of Deusto, Deusto, Spain. Accessed in September 2011. Available at <https://www.weblab.deusto.es/weblab>.
- [5] Open Collaborative Environment for the Leverage of Online Engineering instrumentation, (OCELOT), École d'Ingénieurs Télécom St. Etienne, Université Jean Monnet, France. Accessed in September 2011. Available at <http://ocelot.telecom-st-etienne.fr/>.
- [6] Labshare, The Labshare Institute, Sidney, Australia. Accessed in September 2011. Available at <http://www.labshare.edu.au/>.
- [7] Luís Gomes and Seta Bogosyan. "Current Trends in Remote Laboratories". IEEE Transactions on Industrial Electronics, Vol. 56, No. 12, December 2009. <http://dx.doi.org/10.1109/TIE.2009.2033293>
- [8] F. J. Rodríguez, C. Girón and E. J. Bueno. "Remote laboratory for experimentation with multilevel power converters". IEEE Transactions on Industrial Electronics, Vol. 56, No. 7, July 2009.
- [9] V. F. Pires, J. F. Martins and T. G. Amaral. "Web Based Teaching of Electrical Drives Using a Mechanical Load Simulator". Proceedings of the 34th IEEE IECON, November 2008.
- [10] C. Ferrater-Simon, L. Molas-Balada, O. Gomis-Bellmunt, N. Lorenzo-Martinez, O. Bayo-Puxan, and R. Villafafila-Robles. "A remote laboratory platform for electrical drive control using programmable logic controllers". IEEE Transactions on Industrial Electronics, Vol. 52, No. 3, August 2009.
- [11] The PHP Group. PHP: Hypertext Preprocessor. Accessed in June 2011. Available at <http://www.php.net/>.
- [12] Scott Loban. "Language Choices for Client-Side Scripting", November 2001. Available at <http://www.peachpit.com/articles/article.aspx?p=24266>.
- [13] W3C. HTML & CSS. Accessed in June 2011. Available at <http://www.w3.org/standards/webdesign/htmlcss.html>.
- [14] W3C. Scripting and AJAX. Accessed in June 2011. Available at <http://www.w3.org/standards/webdesign/script>.

- [15] Apache. The Apache HTTP Server Project. Accessed in June 2011. Available at <http://httpd.apache.org/>.
- [16] Oracle. MySQL. Accessed in June 2011. Available at <http://www.mysql.com/>.
- [17] PhpMyAdmin Development Team, phpMyAdmin. Accessed in June 2011. Available at <http://www.phpmyadmin.net/>.
- [18] W3C. XML Schema. Accessed in June 2011. Available at <http://www.w3.org/standards/xml/schema>.
- [19] WampServer. Overview of WampServer. Accessed in June 2011. Available at <http://www.wampserver.com/en/presentation.php>.
- [20] Drew Philips, Secureimage PHP Captcha. Accessed in June 2011. Available at <http://www.phpcaptcha.org/>.
- [21] PHP Classes Repository, PHP SMNP Class. Accessed in June 2011. Available at <http://www.phpclasses.org/>.
- [22] Bill Moran. Phppdflib. Accessed in June 2011, Available at <http://www.potentialtech.com/cms/node/6>.
- [23] The jQuery Project. jQuery: The Write Less, Do More, JavaScript Library. Accessed in June 2011. Available at <http://jquery.com/>.
- [24] GoogleCode. The Google Geocoding API. Accessed in June 2011. Available at <http://code.google.com/intl/nl/apis/maps/documentation/geocoding>.
- [25] Bram Vandamme. Plonk Library. Accessed in June 2011. Available at <http://www.bram.us/>.
- [26] NetBeans. Information about NetBeans. Accessed in June 2011. Available at <http://netbeans.org/>.
- [27] Microsoft. Microsoft Visual Studio 2010. Accessed in June 2011. Available at <http://www.microsoft.com/visualstudio/>.
- [28] Moxiecode Systems, TinyMCE. Accessed in June 2011. Available at <http://tinymce.moxiecode.com/>.

### AUTHORS

**N. Crabeel** was with Instituto Superior de Engenharia, Porto, Portugal as an Erasmus student from the Katholieke Hogeschool Sint-Lieven, Ghent, Belgium (e-mail: [nils.crabeel@gmail.com](mailto:nils.crabeel@gmail.com)).

**B. Campos Neves** is with Instituto Superior de Engenharia, Porto, Portugal where she is an Assistant Professor at the Electrical Engineering Department (e-mail: [bbn@isep.ipp.pt](mailto:bbn@isep.ipp.pt)).

**B. Malheiro** is with the Instituto Superior de Engenharia, Porto, Portugal where she is an Assistant Professor at the Electrical Engineering Department (e-mail: [m bm@isep.ipp.pt](mailto:m bm@isep.ipp.pt)).

This work was supported by Calouste Gulbenkian Foundation, Lisbon, Portugal. It is an extended version of a presentation given during the 1st Experiment@ International Conference, 17/18 November 2011 in Lisbon, Portugal. Manuscript received 12 December 2011. Published as resubmitted by the authors 18 March 2012.