

State-of-the-Art Remote Laboratories for Industrial Electronics Applications

Mohamed Tawfik, Elio Sancristobal, Sergio Martin, Gabriel Diaz, Manuel Castro

Electrical and Computer Engineering Department
Spanish University for Distance Education (UNED)
Madrid, Spain

{mtawfik, elio, smartin, gdiaz, mcastro}@ieec.uned.es

Abstract—There is no doubt that the implementation of practical sessions in engineering education paves the way for students to be familiar with the instruments and thus, with the industrial real-world. Moreover, they augment the learning outcomes by strengthening the understanding of scientific concepts and theories. Unfortunately, there exist a wide gap between the engineering educational curricula and the industrial real-world owing to the lack of experimentation availability. This is due to high cost and administration burdens that have hindered the adoption of practical sessions causing a significant decline in experimentation within engineering education. Recently, with the advent and exploitation of computer and communication technologies, remote laboratories have broadly proliferated among many universities. Remote laboratories provide on-line ubiquitous workbenches unconstrained by neither temporal nor geographical considerations and allow an interactive learning environment that maintains student attention. In this context, hundreds of remote laboratories for industrial electronics applications have been developed and numerous technologies have emerged in order to facilitate their construction and implementation. This paper reports on state-of-art remote laboratories for industrial electronics applications, and their design stages. The paper addresses many solutions in the development and the design stages, along with cutting edge technologies involved. This is in order to foster remote laboratories adoption within industrial electronics disciplines and hence, increase the industrial competencies in engineering education.

Keywords-component; distance education; engineering education; elearning; online learning

I. INTRODUCTION

The criteria of engineering qualification have demanded new skills requirements and competences that match with labor markets. The “learning-by-doing” methodology has a central role in allowing students to strengthen the understanding of the scientific concepts and phenomena, through practical sessions and thus, to be familiar with the instruments and the industrial real-world. Remote laboratories seem to be the best choice in terms of availability, administration and cost. Thereby, recently, a widespread proliferation of remote laboratories, in all industrial electronics disciplines, has occurred at many universities and institutions [1, 2]. However, their design has not received too much attention; most of the published literature focus only on the objectives of a certain application, within a conducted course or on their functional and

operational aspects [3], rather than their design and development. To fill this gap, this paper presents a generic study on state-of-art remote laboratories for industrial electronics applications and their development addressing all the design process stages and cutting edge technologies adopted.

The paper is organized as follows: Section II defines the common generic architecture of remote laboratories for industrial electronic disciplines summarizing the design process stages and the function of each component, Section III reports on the available lab server technologies, Section IV reports on the available client-server communication technologies, and finally a conclusion is derived in section V.

II. COMMON GENERIC ARCHITECTURE

Remote laboratories are those laboratories that can be controlled and administrated online. They differ from the virtual simulated laboratories as they are interacting with physical instruments. The common generic architecture design of today’s remote laboratory for industrial electronics applications could be structured as shown in Figure 1. Next, a comprehensive definition of the main components of this architecture is going to be presented, pointing out the available technologies they mostly rely on.

A. User Interface

The user interface is a virtual end-user workbench that handles all the lab administration process. It is a web site that runs on the user’s web browser and usually requires a server-side programming language to retrieve user’s data from database such as PHP, ASP, and JSP, along with a Graphical User Interface (GUI), which is built by means of plugins and animation technology embedded in the HTML code to resemble the real lab workbench such as Flash, Java Applets and ActiveX controls. Other scripting languages commonly used are: JavaScript; it adds interactive elements to the web page without relying on the server, and AJAX; it retrieves information from the server in an efficient way and without needing to refresh the web page. The website could be supplanted by a software application installed on the user-PC and connected to a database server (application server).



Figure 1. Common Generic Architecture design of today's remote laboratory for industrial electronics applications.

B. Web Server

The web server is a server-PC that hosts the web site and the database files. Apache and Microsoft IIS are the most commonly used servers, while MySQL, Microsoft SQL, and Oracle, are the most commonly used databases. The web server sends the user requests to the lab server in the form of XML messages through TCP/IP model over HTTP layer. Other attempts have been made to use non-HTTP based protocols such as Common Objective Request Broker Architecture (CORBA), Java Remote Method Invocation (RMI), .NET Remoting, and TCP/IP sockets. Some of these technologies, however, are restricted to local networks only [4].

C. Lab Server

The lab server is a server-PC that hosts the instrumentation control software (lab server software) and it is connected directly to the instruments and the controller. The lab server software sends commands to the controller regarding the received requests or the programmed code from the user. The lab server software could be built from scratch with a multi-purpose programming language such as C# and C/C++, or with graphical programming environment such as LabVIEW [5] and MATLAB [6]. Or else, it could be a proprietary software that comes with the controller. The instrumentation control software is connected to the controller and the instruments by standards such as USB, RS-232, Ethernet, General Purpose Interface Bus (GPIB-IEEE-488.2), serial port, parallel port, etc. depending on the controller or the equipment platform. The Data Acquisition Card (DAQ) may be needed to retrieve and convert digital and analog data signals [7, 8]. There are several modular types of instrumentation platforms such as PXI (PCI eXtensions for Instrumentation) [9], LXI (LAN eXtensions for Instrumentation) [10], GPIB [9], and VXI (VME eXtensions for Instrumentation) [10].

D. Controller

The controller is a programmable device that directly controls the controlled objects and they are suited for all types of applications. In the literature, the controllers that have been typically used in remote laboratories are: Programmable Logic

Controller (PLC) [11, 12], Programmable Logic Device (PLD); Field-Programmable Gate Array (FPGA) [13, 14] and Complex Programmable Logic Device (CPLD) [15], and Microcontroller [16]. The controller connection with the instruments and the controlled objects may entail connectors, converters (e.g., A/D, D/A), I²C-based electronic boards, etc. In some applications a relay switching matrix is used to switch and route the connection between instruments and experiment elements [9, 17, 18]. Whilst, some experiments do not require a controller, they only rely on measurements and require a measurement card such as Digital Signal Processing (DSP) card in case of real-time image processing experiments [19]. Otherwise, LabVIEW in conjunction with commercial measurement and experimentation platforms such as NI ELVIS II [20], VISIR [9, 21], and CompactRIO [22] has facilitated the adoption of remote laboratories at many universities.

III. LAB SERVER SOFTWARE

The common outstanding technologies applied for remote laboratories lab server software development are LabVIEW and MATLAB. LabVIEW is an industrial leader graphical programming environment for developing, testing, controlling and monitoring systems using intuitive graphical icons, known as Virtual Instruments (VIs), which imitate the physical instruments. LabVIEW provides a built-in web server for publishing the web pages that allows its frontal panel of VIs to be embedded and directly controlled from a web browser. With DataSocket, LabVIEW allows data exchange between multiple users and applications efficiently over the internet without the complexity of low-level TCP programming. In the literature a wide set of remote laboratories application have been developed using the LabVIEW along with its built-in web server as seen in [8, 11, 23-27]. The drawback is that web publishing tools require that users download and install the LabVIEW run-time engine.

MATLAB is a high-level technical computing language and interactive environment for high performance intensive numerical computation, algorithm development, data visualization, and data analysis. MATLAB enables the creation of applications that use capabilities of the World Wide Web (WWW) by offering additional toolboxes that provide several

possibilities for interoperability and data exchange with other GUI applications. The user sends input parameters of a particular test by the GUI (or by uploading an M-code). The Real-Time Windows Target Toolbox extracts the input parameters and updates the experiment Simulink blocks model. Afterwards, it compiles the model into ANSI C code and executes the code in the lab server PC while interfacing to real hardware using PC I/O boards. Thus, the users do not need to have locally installed MATLAB or to basically know anything about MATLAB programming. Remote laboratories based on MATLAB can be found in [7, 28-30].

LabVIEW and MATLAB possess rich and powerful features to ease laboratories construction, among them are: data exchange with other GUI applications such as Component Object Model (COM), ActiveX, Common Gateway Interface (CGI), Java and .Net applications, and web services; support for standard Application Programming Interfaces (APIs) such as Interchangeable Virtual Instruments (IVI) and Virtual Instrument Software Architecture (VISA) to provide interface-independent communication with different platforms such as GPIB, PXI, VXI, USB, LXI, and others; connection with Open Database Connectivity (ODBC) or Object Linking and Embedding Database (OLEDB) compliance database; compilation as Dynamic Link Library (DLL) files to be called from the Lab server software as a driver to execute the experiments on the hardware; and support for OLE for Process Control (OPC) Servers to enable developing Human-Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) [11, 31].

LabVIEW and MATLAB are widely used in engineering field. They have respective advantages at different aspect. LabVIEW is the most popular remote laboratory environment and it is the most outstanding representative of graphical programming language visualization and parameter tuning for remote operation, while MATLAB is the most powerful computing language for control algorithm development and simulation. Full using of their advantages can achieve high efficiency programming. In numerous remote laboratories applications, this hybrid method was adopted in [32-34] where the signal acquisition and the GUI were developed with LabVIEW, while numerical calculation and signal processing were developed with MATLAB. LabVIEW can support several ways of hybrid programming with MATLAB by means of:

1. *ActiveX automation technology*: directly call MATLAB ActiveX server.
2. *DLL technology*: MATLAB files can be compiled as a DLL, and then called in LabVIEW.
3. *COM technology*: MATLAB files can be compiled as a COM format, and then called in LabVIEW.
4. *MathScript RT Module*: it is a module provided by LabVIEW that provides a native compiler for .m files without requiring additional third-party software and allows embedding .m file scripts in LabVIEW applications by connecting the text-based I/O variables with the inputs and outputs of LabVIEW as shown in Figure 2.

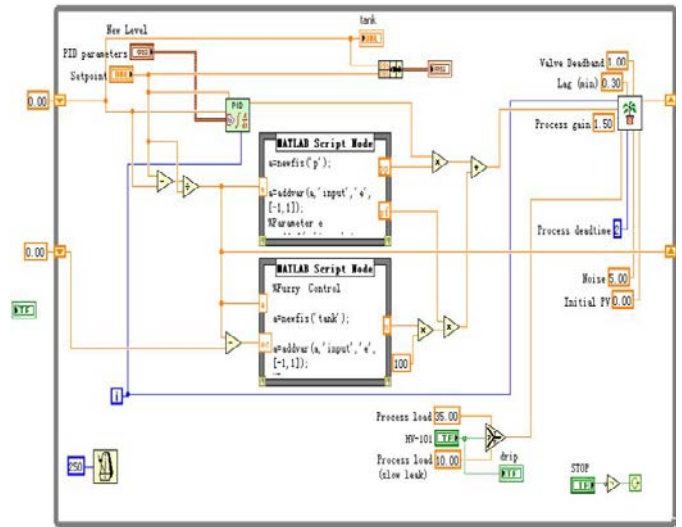


Figure 2. MATLAB m-files embedded in LabVIEW block diagram by MATLAB Script Node.

Else, lab-server software could be developed from scratch by a multi-purpose programming language such as C# and C/C++ as in [12, 35]. In addition, the hardware drivers could be built by LabVIEW or MATLAB in form of DLLs and called from the multi-purpose programming language. The major disadvantage of this approach is the complexity and the necessity of professional programmers.

IV. CLIENT-SERVER COMMUNICATION

As seen previously, there exist many technologies for developing the user interface software and the lab server software. However, the critical issue is to find a suitable solution for their communication between each other. The commonly applied technologies for this purpose are the follows.

A. Desktop Sharing

Desktop sharing allows remote controlling of another computer from the local one either by Remote Desktop software viewer (based on TCP) or by a web page (based on HTTP server) without the need of pre-installed software or plug-in. When the user logs on the PC, he can access to the windows desktop with the maximum restrictions, he can only start the programs associated to the control of the workstation but he can't modify the configuration, shutdowns or viewing the folders in its hard disk. This solution is bandwidth intensive and lacks security. However some features are offered for protection including data encryption and password security. An extra security layer with stronger encryption could be added by tunneling over Secure Shell (SSH) or Virtual Private Network (VPN) connection. For remote laboratories applications, the user access must be controlled by a system hour reservation to support the simultaneous remote access of students. The typically adopted desktop sharing software for remote laboratories are:

- *Virtual Network Computing (VNC)*: it is a cross-platform protocol that is based on image transferring which slow its speed performance. It allows multi-connection but within the same session, so all users see the same thing. Examples on VNC applications are found in [36-38].
- *Remote Desktop Protocol (RDP)*: it is developed by Microsoft and only runs on Windows Operating Systems (OSs). It is based on data transferring, which increase its speed performance rather than image transferring. RDP treats each connection as a new session but it is limited to a single connection. File transfer between the local and the remote controlled machines is not supports as in other desktop sharing software. Examples on RDP applications are found in [39-41].

B. Common Gateway Interface (CGI)

CGI is a standard programming interface that defines the communication over HTTP between web servers and external programs either from an HTML form or by directly typing the Uniform Resource Locator (URL) in the browser. This involves passing information to programs written in any programming language such as Perl, C/C++, and LabVIEW. The main body of software runs on server and only input and output forms are sent across the Web and no additional components or are required. CGI works on all browsers and interacts with any operating system. A CGI script on a web server is a potential security risk because it can provide web access by unauthorized person to any information or service. Another security risk is when you write or upload a bad or insecure CGI script to a web server. This could be mitigated by avoiding unsafe programming practices by writing secure CGI scripts, using hidden form files or data encryption. CGI programs are slow since they need to require a new process for every HTTP request and the database connection must be reopened for the next instance, which cause over heading and excessive pressure or loads on the server. One solution is to use FastCGI[42] or Simple CGI (SCGI) instead, they allow a single, long-running process to handle more than one user request. Remote laboratories applications based on CGI are found in [7, 43].

C. ActiveX and Java Applets

ActiveX controls officially operate only with Microsoft's Internet Explorer and the Microsoft Windows OSs. Programmers can write ActiveX controls in any language which supports COM component development, including C/C++, .NET, Java, and LabVIEW. ActiveX controls are Microsoft Win32 components; they have full access to the operating system, with all the risk that this implies. Microsoft has implemented a registry of digital signatures to verify ActiveX controls. Signed ActiveX controls are embedded with an encrypted digital signature from the software publisher. This signature can be used to verify the source and integrity of the code it is attached to. Regardless of the presence of a signature, users must be sure they know what actions or features the control is intended to perform, and that they have a degree of trust in the purported publisher before allowing the control to

run on their system. ActiveX controls are commonly used with LabVIEW built-in web server applications in order to embed its front panel or to add interactive contents to it as shown in Figure 3.

Java Applets is written using the Java programming language and translated into byte code that is platform and web browser- independent. The byte code can be loaded and compiled at runtime by the clients and then executed using a Java Virtual Machine (JVM). The JVM will need to restart each time the browser starts afresh. Some applets require a later version of Java Runtime Environment (JRE), which will force the client to wait for a large download. The Java programming language and tools were designed with security in mind. Each Java application or applet can be given customized permissions to control what actions it is allowed to perform. Unsigned applets run within their own protected area in memory, which is known as a sandbox. The sandbox model prevents applets from interfering with each other or other processes on the system. Thus, security restrictions may make it difficult or even impossible for an untrusted applet to achieve the desired goals. Users or the JVM can grant signed applets permission to operate beyond the bounds of the sandbox. In addition to the sandbox concept, Java includes robust security libraries that provide encryption, certificate implementation, code signing, permissions, and authentication. A wide set of remote laboratories applications based on Java Applets are found in [3, 12, 35, 44-46].

Java Applets and ActiveX controls, are similar technologies that are based on a downloadable code from the web server to be executed on the local computer in order to provide a rich dynamic content functionality to the browser and to add interactivity to web pages either by embedding the contents or running them on a separate window. However, both use different approaches to provide a similar functionality. Java Applets and ActiveX run on client and hence, do not overload the server. Java Applets contain 8-bit byte code, while ActiveX components contain full 32-bit native code. This is one more reason why Java is significantly slower than ActiveX components. ActiveX components are only downloaded the first time they are accessed and each time a new version of the software is updated. On the other hand, Java applets are downloaded every time they are accessed, which cause a potential load on the network.

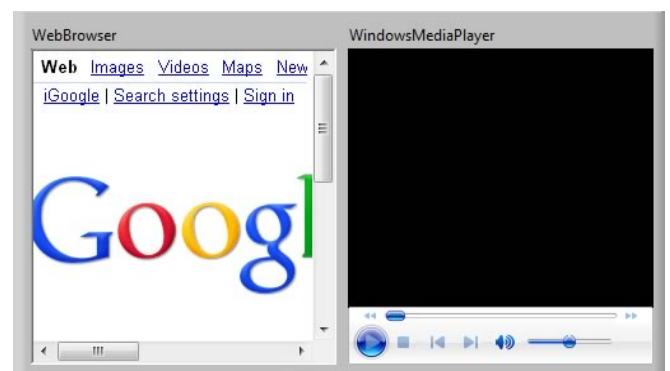


Figure 3. Embedding windows media player and web browser in LabVIEW with ActiveX controls.

D. Rich Internet Applications (RIAs)

RIAs integrate multimedia, graphics, animations and interactivity into webpages, features that are not found in the above mentioned solutions. They are plug-in-based and once the plugin is downloaded, it does not need to be downloaded every time the page is displayed; this reduces application load time, bandwidth requirements, and server load. For security purposes, RIAs run their client portions within a special isolated area of the client desktop called a sandbox. The sandbox limits the visibility and access to the local machine. Adobe Flash is the commonly used RIA in the development of remote laboratories. Adobe Flash is more compatible with all OSs and web browsers. Applications that have adopted Adobe Flash are found in [9, 24, 30, 41, 47]. When it comes to mobile applications, Adobe Flash has shown some drawbacks addressed in [48]. Other competitors RIAs are Microsoft Silverlight and Oracle JavaFX. However, there is no significant adoption of these technologies in remote laboratories applications yet.

E. Asynchronous JavaScript and XML (AJAX)

Recently, several approaches endeavored to substitute plug-in solutions with common web standards such as AJAX and JavaScript [4, 38, 49]. AJAX is a programming technique to build rich and dynamic web applications that quickly respond to user requests. AJAX enables JavaScript to communicate directly with the web server using the XMLHttpRequest object, which requests and updates only the required data on the web page instead of reloading the entire web page. AJAX uses standard technologies that all Web browsers already support, except XMLHttpRequest, such as HTTP, JavaScript, XHTML, XML, Document Object Model (DOM), and Cascading Style Sheet (CSS) to request, retrieve, convert, and present the data on the web page. All recent Internet browsers and platforms support Ajax and there is no need to install any software on the client machine. The major drawback is that AJAX itself does not provide video nor audio capabilities, which is required in high-resolution applications [4], and it does not support the persistent HTTP streaming very well as RIAs, which may be necessary for the real-time data acquisition [50]. Thus, at this moment the use of plug-ins could be the best choice.

V. CONCLUSION

Remote laboratories design is a novice field of science that is gaining everyday more attention. In this paper, a wide angle view on all the design stages in remote laboratories development for industrial electronics education disciplines has been provided. A brief description on the emerging technologies adopted has been presented highlighting features and limitations, along with several literature references to developed applications at many universities. The goal is to foster remote laboratories adoption in industrial electronics education disciplines and to enhance engineering curricula putting it on the line with practical competencies and industrial real-world.

ACKNOWLEDGMENT

Authors would like to acknowledge the support of the following projects: s-Labs (TIN2008-06083-C03-01), SOLITE (CYTED-508AC0341), e-Madrid (S2009/TIC-1650), RIPLECS (517836-LLP-1-2011-1-ES-ERASMUS-ESMO), and PAC (517742-LLP-1-2011-1-BG-ERASMUS-ECUE). As well, Authors would like to acknowledge the support of the VISIR Community and the GOLC consortium.

REFERENCES

- [1] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 4744-4756, 2009.
- [2] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the Art About Remote Laboratories Paradigms – Foundations of Ongoing Mutations," *International Journal of Online Engineering (iJOE)*, vol. 4, pp. 19-25, 2008.
- [3] E. G. Guimaraes, E. Cardozo, D. H. Moraes, and P. R. Coelho, "Design and Implementation Issues for Modern Remote Laboratories," *Learning Technologies, IEEE Transactions on*, vol. 4, pp. 149-161, 2011.
- [4] J. Garcia-Zubia, P. Orduna, D. Lopez-de-Ipina, and G. R. Alves, "Addressing Software Impact in the Design of Remote Laboratories," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 4757-4767, 2009.
- [5] NI LabVIEW - Improving the Productivity of Engineers and Scientists, Retrieved December 30, 2011, from: <http://www.ni.com/labview/>.
- [6] MATLAB - The Language Of Technical Computing, Retrieved December 30, 2011, from: <http://www.mathworks.es/products/matlab/index.html>.
- [7] E. Irmak, R. Bayindir, I. Colak, and M. Soysal, "A remote laboratory experiment for 4-quadrant control of a DC motor," *Computer Applications in Engineering Education*, vol. 19, pp. 747-758, 2011.
- [8] A. Yazidi, H. Henaou, G. A. Capolino, F. Betin, and F. Filippetti, "A Web-Based Remote Laboratory for Monitoring and Diagnosis of AC Electrical Machines," *Industrial Electronics, IEEE Transactions on*, vol. 58, pp. 4950-4959, 2011.
- [9] I. Gustavsson, K. Nilsson, J. Zackrisson, J. Garcia-Zubia, U. Hernandez-Jayo, A. Nafalski, Z. Nedic, O. Gol, J. Machotka, M. I. Pettersson, T. Lago, and L. Hkansson, "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 263-274, 2009.
- [10] U. Hernández-Jayo and J. García-Zubía, "A Remote and Reconfigurable Analog Electronics Laboratory based on IVI an LXI Technologies," presented at the REV 2011 Proceedings, Brasov, Romania, 2011.
- [11] Z. Aydogmus and O. Aydogmus, "A Web-Based Remote Access Laboratory Using SCADA," *Education, IEEE Transactions on*, vol. 52, pp. 126-132, 2009.
- [12] M. Domínguez, J. J. Fuertes, M. A. Prada, S. Alonso, and A. Morán, "Remote laboratory of a quadruple tank process for learning in control engineering using different industrial controllers," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2011.
- [13] J. d. J. Rangel-Magdaleno, J. R. Rivera-Guillen, R. d. J. Romero-Troncoso, and R. A. Osornio-Rios, "FPGA-Matlab-based open core for three-time controllers in automatic control applications," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2011.
- [14] D. Ursutiu, M. Ghercioiu, C. Samoila, and P. Cotfas, "FPGA LabVIEW Programming, Monitoring and Remote Control," *International Journal of Online Engineering (iJOE)*, vol. 5, 2009.
- [15] J. Garcia-Zubia, P. Orduna, I. Angulo, U. Hernandez, O. Dziabenko, D. Lopez-Ipina, and L. Rodriguez-Gil, "Application and user perceptions of using the WebLab-Deusto-PLD in technical education," in *Global Online Laboratory Consortium Remote Laboratories Workshop (GOLC), 2011 First*, 2011, pp. 1-6.
- [16] P. Zenzerovic and V. Sucic, "Remote laboratory for microcontroller systems design," in *MIPRO, 2011 Proceedings of the 34th International Convention*, 2011, pp. 1685-1688.

- [17] M. J. Callaghan, J. Harkin, T. M. McGinnity, and L. P. Maguire, "Intelligent User Support in Autonomous Remote Experimentation Environments," *Industrial Electronics, IEEE Transactions on*, vol. 55, pp. 2355-2367, 2008.
- [18] N. Sousa, G. R. Alves, and M. G. Gericota, "An Integrated Reusable Remote Laboratory to Complement Electronics Teaching," *Learning Technologies, IEEE Transactions on*, vol. 3, pp. 265-271, 2010.
- [19] A. Kalantzopoulos, D. Markonis, and E. Zigoris, "A Remote Laboratory for Real-Time Digital Image Processing on Embedded Systems," *International Journal of Online Engineering (iJOE)*, vol. 5, pp. 25-29, 2009.
- [20] P. Nakazinga, P. Karumuna, A. T. Asimwe, C. Mwikirize, P. I. Musasizi, S. S. Tickodri-Togboa, J. Butime, and A. Katumba, "An Interactive Control Systems Online Laboratory Utilizing NI ELVIS IITM and the Quanser 010 DC Motor Control Trainer," presented at the REV 2011 Proceedings, Brasov, Romania, 2011.
- [21] M. Tawfik, E. Sancristobal, S. Martin, C. Gil, P. Losada, G. Diaz, and M. Castro, "Remote laboratories for electrical & electronic subjects in new engineering grades," in *Promotion and Innovation with New Technologies in Engineering Education (FINTDI), 2011*, 2011, pp. 1-6.
- [22] J. Pechousek, R. Prochazka, and M. Juhanak, "Remote monitoring system for CompactRIO based real-time PID controller," presented at the REV 2010, Stockholm, 2010.
- [23] M. Stefanovic, V. Cvijetkovic, M. Matijevic, and V. Simic, "A LabVIEW-based remote laboratory experiments for control engineering education," *Computer Applications in Engineering Education*, vol. 19, pp. 538-549, 2011.
- [24] M. T. Restivo, J. Mendes, A. M. Lopes, C. M. Silva, and F. Chouzal, "A Remote Laboratory in Engineering Measurement," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 4836-4843, 2009.
- [25] M. Abdulwahed and Z. K. Nagy, "Developing the TriLab, a triple access mode (hands-on, virtual, remote) laboratory, of a process control rig using LabVIEW and Joomla," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2010.
- [26] M. Matijević, M. Stefanović, V. Cvjetković, V. Joković, N. Babajić, M. Ravlić, and S. Nestić, "The development and implementation of a thermal process trainer for control and measurement via the Internet," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2011.
- [27] B. Popović, N. Popović, D. Mijić, S. Stankovski, and G. Ostojić, "Remote control of laboratory equipment for basic electronics courses: A LabVIEW-based implementation," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2011.
- [28] O. Peñaloza-Mejía, L. A. Márquez-Martínez, J. Alvarez-Gallegos, and H. J. Estrada-García, "DSP-based real-time platform for remote control of internet-connected systems," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2010.
- [29] A. Tekin, F. Ata, and M. Gökbülüt, "Remote control laboratory for DSP-controlled induction motor drives," *Computer Applications in Engineering Education*, pp. n/a-n/a, 2010.
- [30] C. A. Ramos-Paja, J. M. R. Scarpetta, and L. Martínez-Salamero, "Integrated Learning Platform for Internet-Based Control-Engineering Education," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3284-3296, 2010.
- [31] S. Reynard, O. Gomis-Bellmunt, A. Sudrià-Andreu, O. Boix-Aragonès, and I. Benítez-Pina, "Flexible manufacturing cell SCADA system for educational purposes," *Computer Applications in Engineering Education*, vol. 16, pp. 21-30, 2008.
- [32] D. Hercog, B. Gergic, S. Uran, and K. Jezernik, "A DSP-Based Remote Control Laboratory," *Industrial Electronics, IEEE Transactions on*, vol. 54, pp. 3057-3068, 2007.
- [33] A. Rojko, D. Hercog, and K. Jezernik, "Power Engineering and Motion Control Web Laboratory: Design, Implementation, and Evaluation of Mechatronics Course," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3343-3354, 2010.
- [34] X. Li, Y. Zhang, S. Wang, and G. Zhai, "A Method for Analog Circuits Fault Diagnosis by Neural Network and Virtual Instruments," in *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, 2011, pp. 1-5.
- [35] F. Zeiger, M. Schmidt, and K. Schilling, "Remote Experiments With Mobile-Robot Hardware via Internet at Limited Link Capacity," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 4798-4805, 2009.
- [36] A. G. Vicente, Mu, x00F, I. B. oz, J. L. L. Galilea, and P. A. R. del Toro, "Remote Automation Laboratory Using a Cluster of Virtual Machines," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3276-3283, 2010.
- [37] S. C. Kong, Y. Y. Yeung, and X. Q. Wu, "An experience of teaching for learning by observation: Remote-controlled experiments on electrical circuits," *Computers & Education*, vol. 52, pp. 702-717, 2009.
- [38] D. Lowe, S. Murray, E. Lindsay, and L. Dikai, "Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 289-294, 2009.
- [39] D. Cmuk, T. Mutapcic, and I. Bilic, "MIRACLE-- Model for Integration of Remote Laboratories in Courses that Use Laboratory and e-Learning Systems," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 275-288, 2009.
- [40] S. R. Jernigan, Y. Fahmy, and G. D. Buckner, "Implementing a Remote Laboratory Experience Into a Joint Engineering Degree Program: Aerodynamic Levitation of a Beach Ball," *Education, IEEE Transactions on*, vol. 52, pp. 205-213, 2009.
- [41] M. A. Bochicchio and A. Longo, "Hands-On Remote Labs: Collaborative Web Laboratories as a Case Study for IT Engineering Classes," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 320-330, 2009.
- [42] FastCGI | FastCGI -, Retrieved January 1, 2012, from: <http://www.fastcgi.com/drupal/>.
- [43] B. Hanson, P. Culmer, J. Gallagher, K. Page, E. Read, A. Weightman, and M. Levesley, "ReLOAD: Real Laboratories Operated at a Distance," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 331-341, 2009.
- [44] N. Lewis, M. Billaud, D. Geoffroy, P. Cazenave, and T. Zimmer, "A Distance Measurement Platform Dedicated to Electrical Engineering," *Learning Technologies, IEEE Transactions on*, vol. 2, pp. 312-319, 2009.
- [45] D. Lopez, R. Cedazo, F. M. Sanchez, and J. M. Sebastian, "Ciclope Robot: Web-Based System to Remote Program an Embedded Real-Time System," *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 4791-4797, 2009.
- [46] E. Fabregas, G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre, "Developing a remote laboratory for engineering education," *Computers & Education*, vol. 57, pp. 1686-1697, 2011.
- [47] Q. Yuliang, L. Guo-Ping, Z. Geng, and H. Wenshan, "NCSLab: A Web-Based Global-Scale Control Laboratory With Rich Interactive Features," *Industrial Electronics, IEEE Transactions on*, vol. 57, pp. 3253-3265, 2010.
- [48] Thoughts on Flash, Retrieved January 1, 2012, from: <http://www.apple.com/hotnews/thoughts-on-flash/>.
- [49] S. Dutta, S. Prakash, D. Estrada, and E. Pop, "A Web Service and Interface for Remote Electronic Device Characterization," *Education, IEEE Transactions on*, vol. 54, pp. 646-651, 2011.
- [50] H. Wenshan, L. Guo-Ping, D. Rees, and Q. Yuliang, "Design and Implementation of Web-Based Control Laboratory for Test Rigs in Geographically Diverse Locations," *Industrial Electronics, IEEE Transactions on*, vol. 55, pp. 2343-2354, 2008.