

Virtual and Remote Laboratories in Process of Control Education

<http://dx.doi.org/10.3991/ijoe.v8i1.1830>

M. Kalúz, E. Čirka and M. Fikar

Slovak University of Technology in Bratislava (STU), Bratislava, Slovakia

Abstract—This paper describes the progress in the development of virtual and remote laboratories at the Department of Information Engineering and Process Control, FCFT STU in Bratislava. This article is the overview of technical solutions used for online education purposes. These comprise software technologies, most commonly used in virtual and remote laboratory design at our department, such as MATLAB Web Server, Java, C/C++, and Adobe Flash. We have created virtual laboratories as online Web applications which provide features of mathematical computing and simulations of technological plants. We also describe the technology of remote control laboratory with a real experimental device.

Index Terms—virtual, remote, laboratory, e-learning

I. INTRODUCTION

Nowadays, in the field of automation and process control it is essential to keep abreast of information technologies. Virtual laboratories became a very popular way how to improve education opportunities in the fields of science.

At our department we try to give students practical experience with measurements using real technological devices. But it is not possible to provide them these opportunities at any time since our capabilities depend on disponibility of real plants at our department. In the past we decided to develop internet virtual laboratories to handle this issue.

Term of *virtual laboratory* can be understood as computer application or program, which is designed for purposes, such as computations of mathematical problems and model simulations. Virtual laboratories can focus on wide fields of science, like biology, chemistry, physics, mathematics, automation, and others. In many cases the best way to provide students and other interested users with the possibilities of virtual laboratories is to publish these applications or programs on the Internet. Unlike offline virtual laboratories, the online alternatives have a great advantage from the accessibility point of view. If they are publicly accessible, users can perform their experiments from home or any other place with Internet connection. For online solution, there are two main most commonly used modes of operation. The first is client side application, which is downloaded after execution to client and uses program resources of host computer. Such technologies are represented, for example, by HTML+JavaScript, Java Applets, Adobe Flash Applications, etc. The second type is client-server solution, which uses program resources from server computer. This type of solution needs a more complex implementation but also provides better features and computation potential. Client-

server solution can be built on technologies like HTML+PHP/ASP, Java Servlets, MATLAB-related technologies, etc.

In the case when virtual laboratory is designed for simulation of mathematical models of technological plants, the results precision closely depend on the accuracy of the model. If we need more precise approach, the complexity of model raises rapidly. This problem can be solved by replacing mathematical model of plant with real technological plant and by redesigning the communication technology between client and server side. This type of solution is called remote laboratory, because user can be located in the different place than experimental device. The remote laboratory can be designed on the basis of wide spectrum of technologies, above mentioned including. The choice of technology is strictly individual and depends on the type of remote laboratory and its usage.

The virtual and remote laboratories from the different fields of science can be found all over the Web. The interesting example of chemistry virtual laboratory was developed by collective from Carnegie Mellon University, Pittsburg, Pennsylvania, USA. They created a Java Applet based online laboratory [1] providing different chemical experiments in user-friendly application with fully interactive graphical user interface.

Software developers from Loughborough University, Leicestershire, UK, came with online laboratory combining features of virtual model simulations and remote control of real experiments [2]. For application design, they used programming tool LabVIEW and for online implementation the open-source content management system (CMS) Joomla.

Another interesting solution of remote control laboratory [3] comes from FEI STU in Bratislava, where remote control software for the thermo-optical device uDAQ28/LT [13] was developed. This solution is based on Java Applet and Java Servlet communication, and MATLAB used as main control software.

Our online laboratories are based on several different technologies and most of them are published in our online educational courses [4] at Moodle [5]. For most of our computation and simulation virtual laboratories, we used technologies connected with MATLAB. Advantages of these types of solutions are that we can use the computation potential of optimized functions of MATLAB, so that we can simplify the development.

For process model simulations in another virtual laboratory we used Adobe Flash programming platform. The Flash was also used as GUI application in our remote laboratory. In comparison with most commonly used technologies for client applications in virtual laboratories,

such as HTML/PHP and Java Applets, the Adobe Flash programming platform is more suitable for graphical design development and also provides more options for graphical user interface customization. From the simplicity point of view and software/hardware requirements, Java and HTML/PHP solutions seem to be better but not so far.

In the last few years, we have developed virtual laboratories based on several technologies. The list of them is as the following:

- MATLAB Web Server Laboratory
- Virtual laboratory based on JavaServer Pages with MATLAB functions
- Virtual laboratory based on Common Gateway Interface (CGI) executables with MATLAB functions
- Virtual laboratory based on Adobe Flash platform

A. MATLAB-related technologies

MATLAB as a strong and effective computation program provides wide possibilities of user script and features deployment to the Web applications. MATLAB Compiler is a tool that can transform regular MATLAB scripts and Simulink model files into DLL and LIB libraries, MS Windows executables, and sources for technologies such as MS Excel, JAVA, COM, and .NET. These technologies replaced the older and discontinued MATLAB Web Server technology.

B. MATLAB Web Server

Our first attempts of publishing MATLAB features on the Web were performed via MATLAB Web Server technology. MATLAB Web Server is client-server technology connecting client side Web GUI (most commonly HTML Web pages) and MATLAB, which is running on server side. Communication is served by HTTP daemon on regular HTTP Web server (Apache). User-defined scripts in form of M-files are located on the server, where they are called through CGI executable matweb.exe and processed by MATLAB. The results of MATLAB Web Server requests are generated in the form of regular HTML Web page. This technology provides also possibility of generating graphical results.

In the past, we have built several virtual laboratories and Web tools based on MATLAB Web Server such as Dynamic Simulations of Chemical Processes [6], Internet Module for Process Modeling and Simulation [7], and Experimental Identification Toolbox – EXPID [8].

After MATLAB Web Server technology was discontinued from version of MATLAB 2006b we decided to focus our attention on MATLAB Compiler and its features.

C. MATLAB & Java

MATLAB Compiler’s Builder JA is a tool component designed for creating Java language classes from user-defined MATLAB functions. MATLAB functions can be transferred by builder to byte code Java classes, which can be used as a part of Java Web applications. M-files including m-code can be compiled into one or more distributable

JAR package files. Application using MATLAB Java code can be described as stand-alone applications. They do not need MATLAB to be run but only MATLAB Compiler Runtime (MCR). Another plus of this technology is that Java is programming language independent on platform. This means that programs and applications created in Java can be run on any operating system using Java Virtual Machine (JVM). The main disadvantage of this approach is that we cannot compile some MATLAB components, such as Simulink models, into Java classes, because MATLAB Builder JA does not provide these features.

We built virtual laboratory using MATLAB functions in a form of Java components, and Java ServerPages (JSP) technology for GUI implementation [9]. Virtual laboratory based on Java and MATLAB contains applications with the following features:

- Polynomial mathematical operations and roots finding (Fig. 1)
- Various matrix operations (Fig. 1)
- Transfer function algebra (parallel/serial, closed-loop with positive/negative feedback) (Fig. 2)
- Pole-zero map, step response and impulse response of LTI models
- Process model simulations including storage tanks and heat exchangers (Fig. 3)

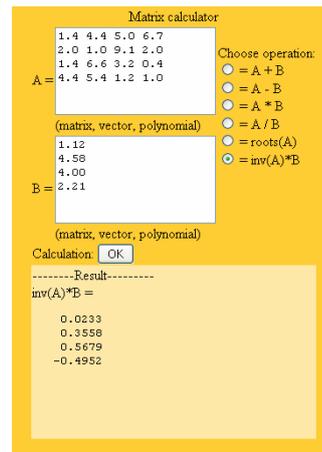


Figure 1. Application for matrix, vector, and polynomial operations

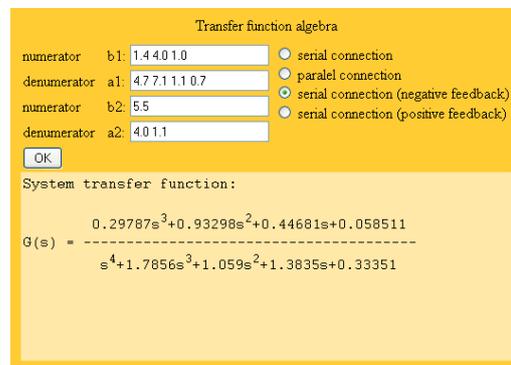


Figure 2. Application designed for mathematical operations of transfer functions in different type of connections

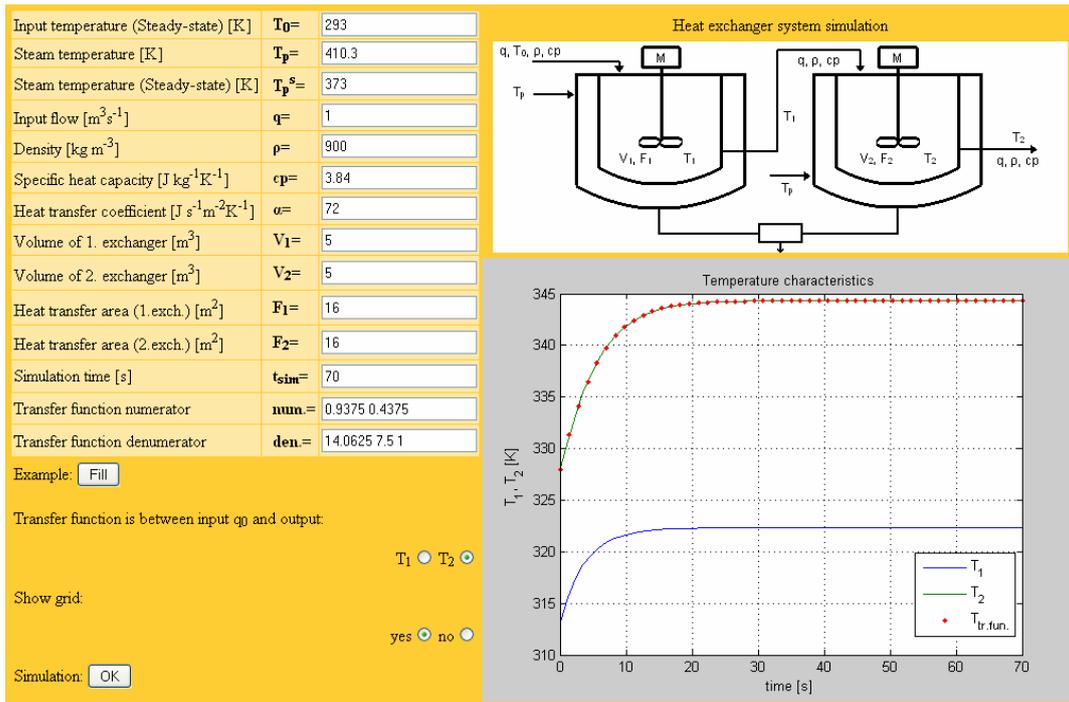


Figure 3. Application for process model simulation (heat exchangers)

D. MATLAB & C language

One of the features of MATLAB Compiler is building Common Gateway Interface executable C/C++ binary files from M-files. These built programs can be described as stand-alone applications, which can be directly executed by operating system. MATLAB Compiler has also features of building CGI binaries from Simulink model files, using Real-Time Workshop. These can be combined with built MATLAB functions and embedded in wider solution to create Web application that provides features of model simulation.

In our attempts to examine more than one MATLAB deployment technology we decided to rebuild Java virtual laboratory (mentioned above) using MATLAB C/C++ binaries. We created a virtual laboratory which uses MATLAB functions and models compiled into C/C++. They are located on the Web server and called through Common Gateway Interface from regular HTML Web forms, which are used as graphical user interfaces.

The list of applications is almost identical as the list in Java virtual laboratory. One of the new features is application for LTI model simulations (Fig. 4).

The application GUI is a regular HTML Web page containing several input fields and control components. User can set the mathematical model in two forms: numerator and denominator of transfer function model, or matrices of state-space representation. There is also an option to compare both types of models in one simulation. The inputs for model simulation can be set as scalars for transfer function and SISO state-space models, or as vectors for MIMO state-space models. User can also set a custom input steps to simulation by constructing the matrix with time vector in the first column and corresponding input steps in the other columns. The results of simulation are displayed as graphs of inputs and outputs with the possibility of graphical customization by setting up the line modifiers, grid and legends.

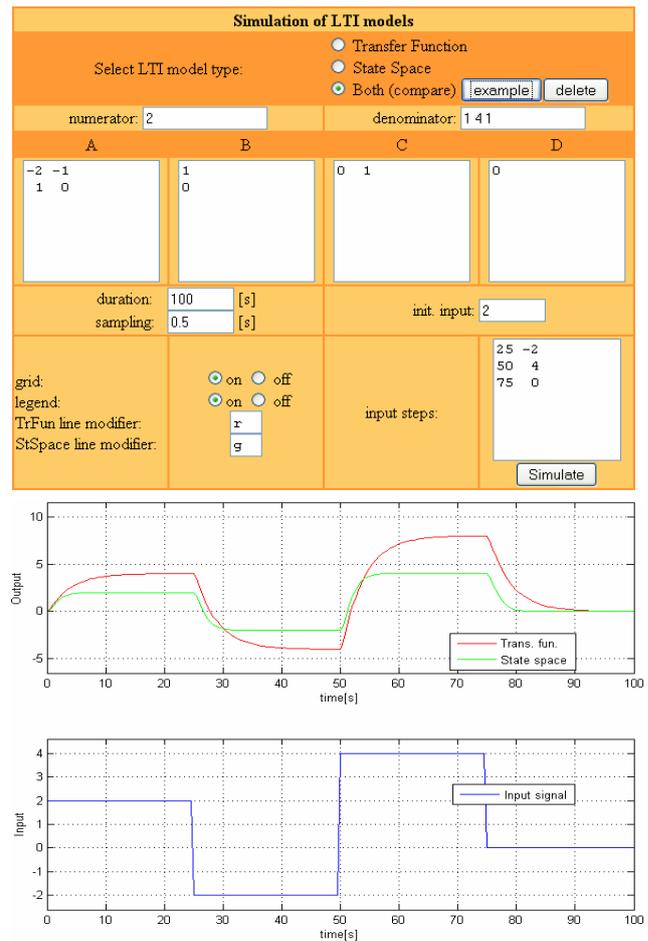


Figure 4. Top figure is showing application for simulations of LTI models using MATLAB C/C++ executables, and communicating through CGI. Bottom figure shows the result graph of LTI model simulation.

E. Simulation laboratory based on Adobe Flash platform

Our virtual Flash laboratory [10, 11] was created using software programming platform Adobe Flash, and it provides virtual simulations of technological plants. The simulation applications were created as Flash SWF files, published on regular HTML Web page. We created three applications, presenting technological plants most commonly used in education process at our department. The first application provides simulation of tank storage system (Fig. 5). It contains two general mathematical models of plant, the non-linear and linear. Each of the models optionally consists of 1-3 ordinary differential equations (ODE) depending on number of tanks in the system. The second simulation contains model of a tube heat exchanger (Fig. 6) which is discretized to five segments. Mathematical model consists of five linear ODEs. The last application is simulation of continuous stirred-tank reactor dynamics (Fig. 7) with mathematical model containing both types of ODEs (linear and non-linear). Lately, we extended all applications with features that provide a possibility of closed-loop PID control simulation. Applications also provide features of data export into several different structures (Fig. 8).

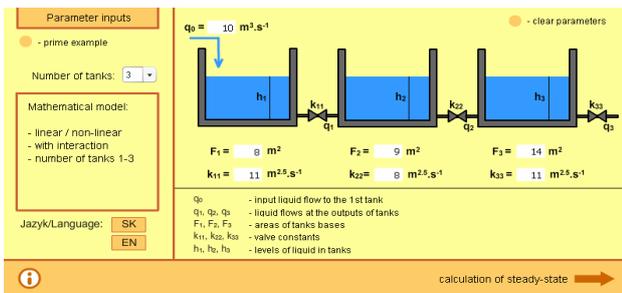


Figure 5. First screen of application for simulation of tank storage system

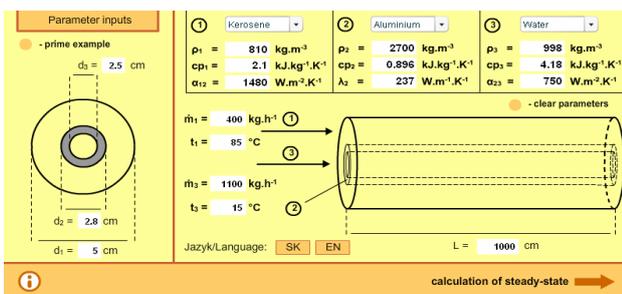


Figure 6. First screen of application for simulation of tube heat exchanger

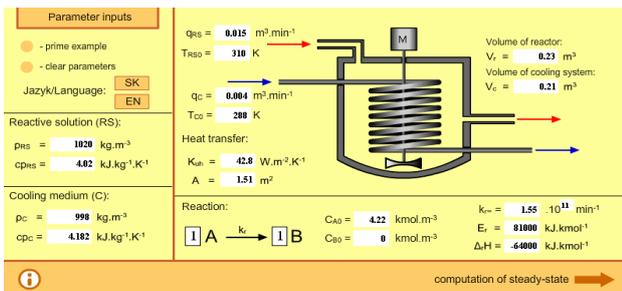


Figure 7. First screen of application for simulation of continuous stirred-tank reactor

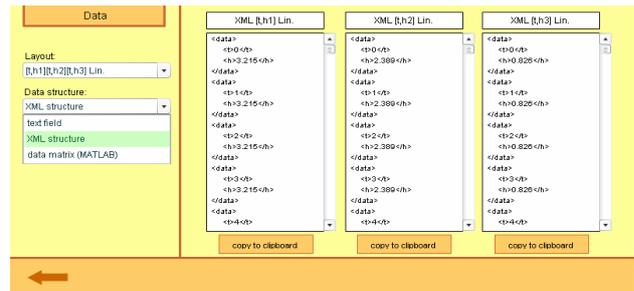


Figure 8. Data export from measurement. User can choose between three types of data structure (plain text, XML, and MATLAB matrix)

II. REMOTE LABORATORY

In a later phase of online education improvement at our department, we developed a remote control laboratory [12]. The laboratory was created using several technologies connected together. Connections between technologies used in our remote laboratory are described in Fig. 9.

We used Adobe Flash platform for client side GUI implementation. At the server, we used PHP for main processing scripts, MySQL database as a communication channel and a data storage system, and MATLAB as main control system for remote device. Client side GUI consists of several screens. The first is a login form where user fills in his name and password to get an access to the second screen where reservation system (Fig. 12) is located. Before user can proceed to measurement, he has to create a new valid reservation for experiment. This feature is designed for avoid time collisions between measurements of different users. On the next screen user can setup his experiment (Fig. 13) by choosing Simulink control scheme and its parameters. When all input parameters are correctly filled in, application will execute MATLAB on server to perform the experiment (Fig. 14). GUI of application also contains a video stream from remote web camera that is aimed at controlled device, so that the remote user can observe the behavior of the experiment in its real time.

For the purposes of remote laboratory and application testing, we chose a thermo-optical device uDAQ28/LT [13] (Fig. 10) created by DigiCon Corporation. It is originally designed for education in the fields of automation and process control. Device can be connected directly to

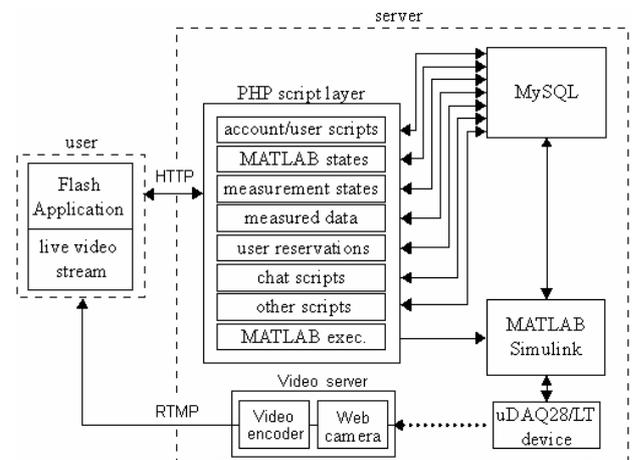


Figure 9. Scheme of remote control laboratory solution

AB.



Figure 10. Experimental device uDAQ28/LT

computer by universal serial bus (USB) and after simple driver installation it can communicate with MATLAB

Remote laboratory contains all default simulation schemes provided by device manufacturer but there is also an option to upload custom scheme created by user. For this purpose we have created upload form that provides all necessary features important for keeping software security and operability. Before a new schema is uploaded to

server and saved to database, the security PHP script checks whether the file has proper extension, size, and whether it contains all necessary parts, such as database connector block and block for data export. In last step PHP script reads the file to indicate the potential danger content (for example system commands written by user).

When experiment is finished, user can download all measured data from database in chosen format. Application can provide data in the form of structured XML file, plain text file or MATLAB m-file. Every experiment result is stored in database with unique identifier, and can be accessed later. Web module for data export is shown in Fig. 11.



Figure 11. Web module for data export

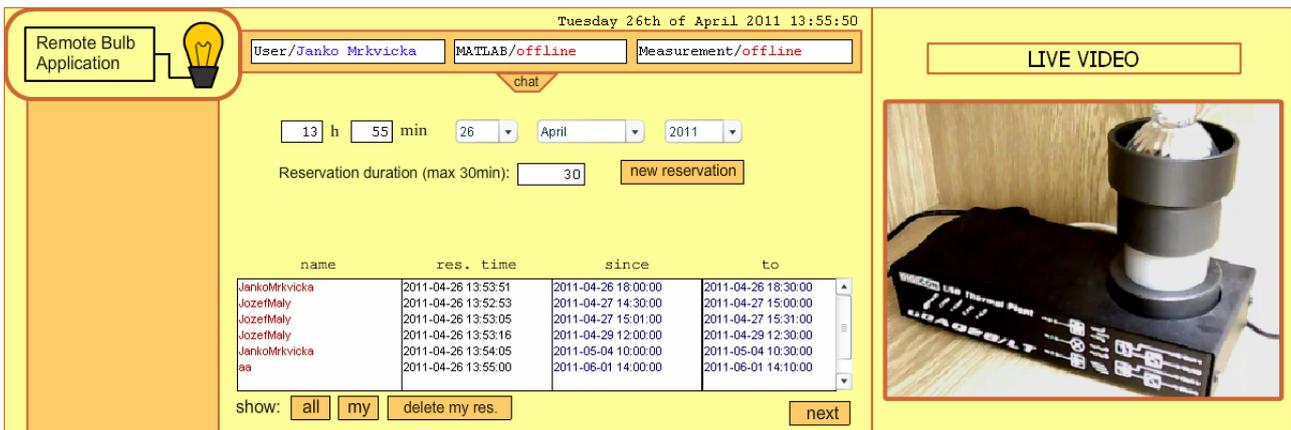


Figure 12. Reservation form for remote experiment

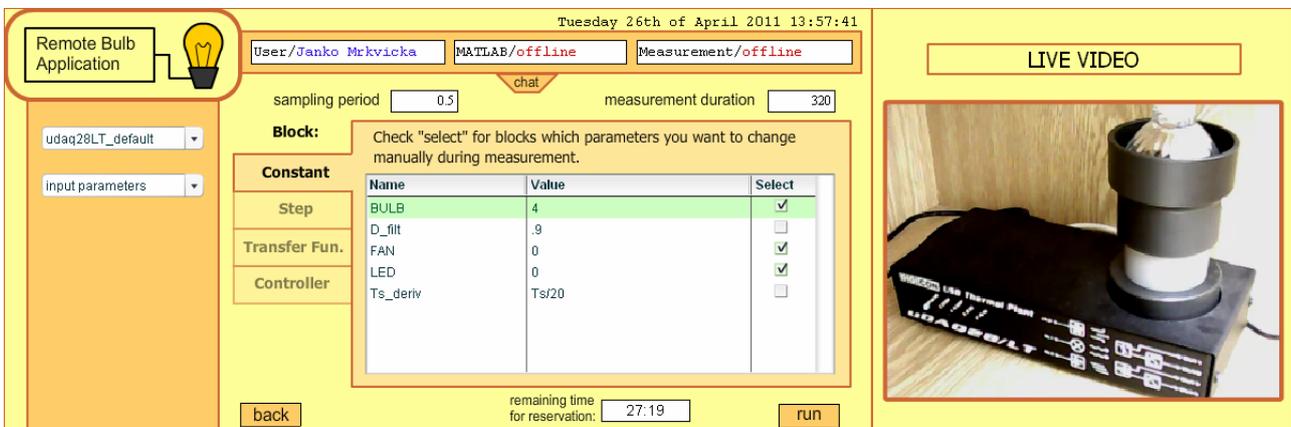


Figure 13. Setup screen where user can select Simulink scheme and its parameters

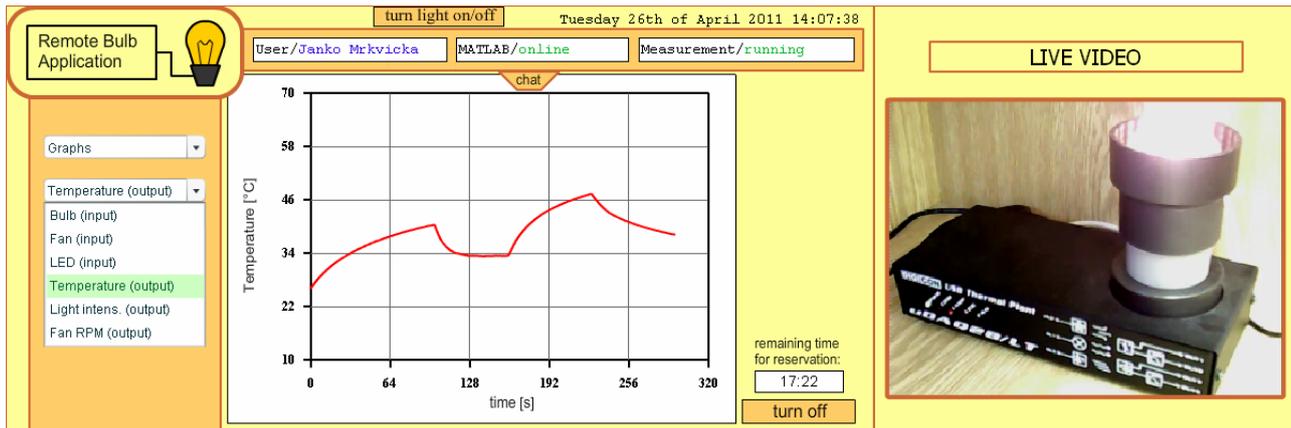


Figure 14. Screen for experiment observation, including graphs of all input and output signals

III. CONCLUSION

Virtual and remote laboratories can be a suitable way how to improve education in the field of automation engineering and process control. Our developed solutions present different ways how education aid tools can be performed. We have chosen several different technologies which can easily handle features that are required to help students and other interested people to gain practical experience. Another benefit of trying wide spectrum of technologies is that we also improve our own skills of software development, so in the future we can create more sophisticated solutions to increase quality of education at out university.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09.

REFERENCES

- [1] D. Yaron, M. Karabinos, D. Lange, J. G. Greeno, and G. Leinhardt, "The ChemCollective: Virtual labs and online activities for introductory chemistry courses", *Science*, Vol. 328, No. 5978, April, 2010. <http://dx.doi.org/10.1126/science.1182435>
- [2] M. Abdulwahed, 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*, <http://dx.doi.org/10.1002/cae.20506>, 2011.
- [3] P. Bisták, and M. Beránek, "Java client server applications for remote laboratory based on matlab and com", 7th International Conference Virtual University VU06, Bratislava, Slovakia, 2006, pp. 135–139.
- [4] M. Bakošová, "E-learning in the Course Integrated Control in the Process Industry", *Proceedings of 10th International Conference Virtual University*, Bratislava, Slovakia, 2009.
- [5] M. Dougiamas, "Moodle – a free, open source course management system for online learning", <http://moodle.org>, 2005.
- [6] L. Čirka, M. Bakošová, M. Fikar, and M. Herceg, "Dynamic Simulations of Chemical Processes via the MATLAB Web Server", *Proceedings of the 15th Annual Conference Technical Computing Prague 2007*, pp. 34–34.

- [7] L. Čirka, M. Bakošová, M. Kvasnica, M. Fikar, "Internet Module for Process Modelling and Simulation", *Principia Cybernetica 2010*, Technical university of Liberec, Liberec, Czech Republic, 2010, pp 22–26.
- [8] P. Dovaľ, L. Čirka, and M. Fikar, "EXPID - Experimental Identification Toolbox", *Proceedings of the 15th Annual Conference Technical Computing Prague 2007*, pp. 38–38.
- [9] M. Kalúz, L. Čirka, and M. Fikar, "MATLAB Builder JA in Control Engineering Education at FCFT STU", *Technical Computing Bratislava 2010*, vol. 18, pp. 053_1–053_5.
- [10] L. Čirka, M. Kalúz, M. Kvasnica, and M. Fikar, "Virtual laboratory", *Proceedings of the 9th International Scientific-Technical Conference Process Control 2010*, University of Pardubice, Kouty nad Desnou, Czech Republic, 2010, pp. C029a-1–C029a-8.
- [11] M. Kalúz, L. Čirka, and M. Fikar, "Virtual Laboratory of Process Control", *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 2011, pp. 348–351.
- [12] M. Kalúz, L. Čirka, and M. Fikar, "Remote Control Software for Thermo-Optical Plant", *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 2011, pp. 587–592.
- [13] M. Huba, P. Kurčík, and M. Kamenský, "Thermo-optical device uDAQ28/LT, Technical and user manual", STU Bratislava, Ilkovičova 3, Bratislava, Slovakia, 2006.

AUTHORS

M. Kalúz is with the Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia (e-mail: martin.kaluz@stuba.sk).

L. Čirka is with the Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia (e-mail: lubos.cirka@stuba.sk).

M. Fikar is with the Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia (e-mail: miroslav.fikar@stuba.sk).

This article is an extended version of a paper presented at the International Conference ICL2011, held in September 2011, in Piešťany, Slovakia. This work was supported in part by the Cultural and Educational Grant Agency of the Slovak Republic. Received 28 September 2011. Published as resubmitted by the authors 9 February 2012.