

SciLab Based Remote Control of Thermo-Optical Plant

<http://dx.doi.org/10.3991/ijoe.v7i4.1783>

M. Jáno and K. Žáková

Slovak University of Technology, Bratislava, Slovakia

Abstract—The paper deals with the web-based implementation of the control system of a thermo-optical plant. The control of the plant is based on the SciLab software which originally is not designed for web-based applications. The paper shows a possible way to circumvent this limitation. The ultimate goal is to enable remote controlled experiment using SciLab. The paper also describes possible tools for communication and control of the real plant and visualization of results.

Index Terms—computer aided engineering, control design, online services, student experiments.

I. INTRODUCTION

Expansion of Internet has provided numerous significant educational opportunities. Web-based teaching materials have been widely produced and used for various educational subjects. This tendency has also influenced the education process in the area of automation. Teaching materials were supplemented by various tools bringing a certain degree of interactivity. Using Internet students can find ever-growing number of interactive examples, animations, and experiments illustrating more or less the topic that is taught.

The development of Internet based control applications can be done in various ways. It is possible to create a completely new application where one has to take care about all steps of the development of a client-server application including building of mathematical mechanism needed for simulation executing. Another possibility is to use support of already developed software that can serve as a simulation engine of virtual or even remote experiment. For this purpose one can consider standard software that is widespread at universities over the whole world such as Matlab, LabView, SciLab, ... The paper is dedicated to the second approach whereby our attention is devoted not to the commercial software but to open and free technologies.

In the last decade the development of the low cost software (free and open source software) takes increasingly bigger and bigger importance. It brings competition to the proprietary software products including open source platforms. Open standards are also supported by European Union that would like to create conditions for the market development and to avoid the constraints given in the case when the only exclusive producer creates software for certain area. The direction was also supported by the speech of European commissioner Neelie Kroes with title “Being open about standards” from June 10, 2008 where she recommends to use the software based on

open standards. This recommendation was headed to governments of all European countries.

The kernel of the presented application engine is formed by SciLab package. SciLab belongs to the software (developed on the base of open and free technologies) that enables to simulate dynamical systems. In its nature it reminds Matlab mainly because of its graphical interface Scicos that is very similar to Simulink.

It is to say that SciLab is usually used for computations that are accomplished on the computer locally. However, the increased expansion of Internet together with the growing support of online education raised a question how to exploit capabilities of SciLab for these purposes, too. In such a case, one installation of SciLab placed on the remote server could serve for several clients whereby the client can be represented by a person or an application.

We started to test SciLab for various online interactive examples and simulations. Later we tried to use it for a remote experiment. Such approach is also presented in this paper.

II. PLANT

The introduced thermo-optical laboratory plant uDAQ28/LT (Fig. 1) presents a system that enables to control two physical variables – the temperature inside a plastic cylinder and the intensity of a light source.

The plant [4] has three inputs – the bulb voltage (the heat & light source), the ventilator voltage (system cooling) and the light diode voltage (the second possible light source). In addition to these 3 input voltages there also exist two parameter inputs for adjusting the sampling period and the time constant of the built in derivative filter.



Figure 1. The combined thermo-optical laboratory plant

The user can use eight measured outputs: the system temperature measured by a PT100 sensor, the light intensity (both measured directly, or with a preliminary filtration by the filter of the 1st order) and its derivative, the ambient temperature, the current and the rotation speed of the ventilator.

The whole plant is supplied by 12V/2A DC external adapter. The advantage is that the whole system can communicate with a computer via a USB interface. No special card is required. Since the USB port can be usually found on all today's computers including notebooks, the introduced equipment can be easily widespread.

The communication with the computer runs via the string exchange. The data transfer rate is 250kbit/s. The plant enables to use a sampling period 40-50 ms whereas considering the dynamics of the presented system 1 second should be sufficient for its quasi-continuous control.

Even though the user can measure 8 variables, students usually do not use all of them. They control the temperature inside the plastic cylinder that is influenced by the bulb heating and the ventilator cooling. In spite of the fact that both these variables can be used for control at the same time together, the plant is very often controlled only by the bulb voltage with the ventilator being considered as a disturbance factor. Another possibility is to control the light intensity that can be influenced by the voltage on the light diode.

III. COMMUNICATION WITH A REAL PLANT

Scicos (a part of SciLab) is software that is primarily determined for process modeling and simulation. Therefore it doesn't contain a possibility to control real plants directly. However, it has several possibilities how this problem can be solved. The communication in real time is dependent both on the chosen solution and on the operating system, as well. Linux solution is supported directly by the OS kernel but OS Windows was not designed as the real time operating system. Therefore the Windows solution can be found only with some constraints.

In this part we try to mention the most frequent possibilities that are used for the real time control in Scicos together with their advantages and disadvantages. Our main interest was to find a solution that could be used with both operating systems, i.e. with OS Linux and also with OS Windows.

A. *Scicos-HIL*

Scicos-HIL [7] (Hardware In the Loop) enables to incorporate blocks representing external devices and input-output cards into Scicos schemes. It enables both digital and analogue communication. It supports more than 100 A/D and D/A cards, communication via serial and parallel port and also via USB interface (both versions 1.1 and 2.0). Communication uses two standards – Comedi for Linux and Measurement Computing determined both for Linux and Windows operating system. The interface standardized for both operating systems is the main advantage of the solution. The disadvantage consists in the fact that in this solution each port of the real device has to be represented by an independent block in a Scicos scheme. In the case that real plant has several ports, the simulation scheme can become

very obscure. The possibility to set block parameters is also very low. It can bring problems when it is necessary to establish communication with a device that accepts both numerical and string values as well.

B. *Scicos-RTAI*

Project RTAI (Real-Time Application Interface for Linux) [2] is an extension of the Linux kernel. It enables to create applications communicating with external devices in a real time. Scicos-RTAI is represented by a group of Scicos blocks that use this extension. Scicos-RTAI contains extended functions for creation of independent executable applications. They can be created in two steps: first of all the C code is generated on the base of Scicos simulation scheme. Then, the C code is compiled to executable application. Scicos-RTAI can also communicate with A/D and D/A cards that use Comedi drivers.

The advantage of this solution is the communication in a real time that can be also used for very short sampling periods. The disadvantage is the strict cohesion only with the Linux operating system and the necessity to interfere with the Linux kernel. Actually, it is not always a desired solution.

C. *C-Code Generator*

Scicos enables to generate C programming code from the block scheme. The code can be later compiled to the executable file that can be used together with previous solutions (Scicos-HIL, Scicos-RTAI). The second possibility is to take the generated code and to supplement it by an external function for communication with a device. The advantage is that the operation of the final application doesn't require SciLab running. There runs only a small program containing the necessary functionality. The disadvantage consists in the fact that Scicos doesn't manage to transform all blocks to C code. Blocks generating events (e.g. timer) cannot be transformed. In addition, the generated C code is not able to communicate with a real device by itself. It always needs some additional problem solution.

D. *External Driver*

Another approach is to create a special external driver for the given plant designed especially for simulation in Scicos. The driver has to be completed with Scicos blocks representing the plant.

Sometimes it can happen that it is possible to use a driver developed for other applications. After it is compiled to a dynamically linked library, it can be imported to Scicos where the real device can be represented by a general generic block. However, this procedure is possible only for very simple drivers that can be used with frequently used devices. More preferable way is to design own driver for the specific plant.

E. *Final Discussion*

All mentioned solutions have their pros-and-cons and they are all verified by sufficiently big user group. Let's summarize disadvantages of all solutions.

Scicos-RTAI is capable to work in hard real time, but it is focused only on Linux operating system. Scicos-HIL is solution working under both Linux and Windows operating system, but it is based only on single port Scicos

blocks. Since both solutions try to be general and to cover the maximum of possible applications, it can happen that the final implementation is quite complicated.

C-code generator requires regular compiling of a simulation scheme in the real service of application and it also requires an external solution for communication with a real device. It means that each designer of a new simulation scheme has also to be a good programmer in the C language. In reality this requirement is very difficult to achieve.

The problem of external driver consists in the fact that there exists only a small amount of devices that are supplied with a driver for SciLab/Scicos environment. To build the own one the user has to have quite good knowledge of the internal functionality of Scicos blocks and internal structure of dynamical libraries.

On the base of these conclusions we decided to build an own external driver for the thermo-optical plant. The reason was that the plant has a simple communication protocol and therefore the creation of the driver was not so difficult. Except of this the driver can be used also with other SciLab solutions, i.e. not only for the solution that is presented in this paper.

IV. REMOTE CONTROL AND VISUALISATION

Firstly, it is necessary to mention that SciLab doesn't contain functions for remote control and for online visualization of results. Our requirement was to enable remote user to run experiment with own simulation parameters, with own control algorithm and to visualize results in a real time. The problem can be solved using additional technologies. Of course, we cannot mention here all solutions but let introduce at least some of them.

A. Java

Java programming language was from its beginning designed as the language independent on the platform with a wide spectrum of functions oriented also to the web application development. Nowadays, Java servlets represent maybe the most frequent Java based approach that is used for the design of web applications. They run on the server side and enable to increase a functionality of a web server. They can cooperate with databases or enable communication via HTTP protocol. Using graphical libraries (e.g. Swing) it is possible to create a platform independent solution that enables unified approach to all elements of web application.

B. XMLlab

XMLlab [8] is a programming environment that enables to create a web interface for realization of experiments without knowledge of more complex programming languages as e.g. C language or SciLab script languages are. Graphical design of the web application and setting of parameters are done on the base of XML language that is wide spread among various types of applications. XMLlab enables to include control of application and visualization of results as a part of web page, that is easily accessible from any web browser. However, at this time the control via web page is implemented only for Linux OS and it also requires other additional applications such as Apache web server or VNC program for remote management of computer. Results can be visualized via an animation, too. The problem is that the animation cannot

be realized in a real time. It can be accomplished only after the computation of all values.

C. LAMP and WAMP Environments

Next possibility is to use LAMP (Linux – operating system, Apache – web server, MySQL - database, PHP – script language) and WAMP (Windows, Apache, MySQL, PHP) program environments with corresponding extensions. The abbreviations come out from names of technologies and programs that are used at the development of web applications. In spite of the fact that PHP script language also includes functions for dynamical generation of figures, it is not an ideal solution when we take into account that the visualization should be done in a real time. However, using some of graphical JavaScript libraries can solve this problem in a quite simple way.

D. Final Discussion

As in the previous paragraph let's try to summarize disadvantages of mentioned solutions.

XMLlab represents an interesting approach but it is oriented mainly to Linux operating system and it doesn't allow visualization of results in a real time.

The solution based on Java language enables real time visualization without problems. In the same way the portability among various platforms doesn't bring any problem. However, Java program needs to be compiled. Since servlets typically embed HTML inside Java code, the later modification of code, its enhancement and bug fixing can be difficult. Considering Java applets they require installing a plug-in that allows their running on a client computer. It can happen that the end user has not such privileges.

Following these facts we decided to use the third solution (LAMP, WAMP packages) with combination of JavaScript library called Flot that will serve for real time visualization of results in the final web application. This approach should not cause any difficulties since an interpreter of JavaScript is a part of each modern web browser.

V. REALISATION

The basic problem that has to be solved at building online experiment is to find a way for communication between a remote client and the technological equipment (Fig.2). Actually, for accomplishing the remote control of experiment it is necessary to create two applications: application for the server and application for the client. For simplicity we can consider that the remote client is a typical Internet user with Internet browser and the technological equipment is the controlled plant – in our case the thermo-optical plant.



Figure 2. Communication between remote client and plant

A. Applied Technologies

We used various technologies for the realization of the whole application.

New Scicos blocks were developed in C programming language. The same language was also used for the development of driver for the thermo-optical plant.

The web portal is created using HTML mark-up language in combination with PHP scripting language that is used for generating of dynamical parts of the web page. The graphical layout is customized using cascade styles.

JavaScript ensures interaction with the web page environment on the client side. For this purpose we use mainly the jQuery library that facilitates work with web page elements. Recently the library is quite popular among web designers because of its simplicity and possibility to increase its functionality by various extensions. As it was already mentioned the graphical visualization of variables is done using JavaScript library called Flot that is also built on the base of jQuery.

The server side of the created client-server application that ensures interaction between the web portal and the Scicos environment is realized by Python programming language. Python is an interactive interpreted language with wide possibilities of implementation. The language was chosen because of its simple syntax, fast development, and ready functions for all used technologies.

B. Driver

The driver is in fact a set of functions (in our case written in C language) for setting of data transfer rate, device initialization, data reading, data writing and closing communication with the real device. The created driver can be used not only for the solution in SciLab but also for other solutions outside this software environment.

The communication was ensured using the LibUSB library [3]. It is an open source library written also in C language representing an interface between an application and a real device connected to USB port. It is distributed under the LGPL license. It was developed mainly for Linux operating system but there also exists a version for Windows. Since these two versions are very similar we were able to create a driver for both operating systems without any essential change.

C. Scicos Block for the Plant

Since the simulation scheme for the experiment will be created in the Scicos environment, it was necessary to develop a new block representing a thermo-optical plant. The name of this block is uDAQ28/LT (Fig.3).

The realization of the block consists of two steps:

- It was necessary to build an executive part of the block that has to ensure communication with the created driver (this functionality was realized in C language).
- The block has to have some graphical representation that is visualized in the Scicos block scheme. The graphical form (realized in SciLab script language) defines e.g. number and type of input and output ports, information about block parameters, predefined values, block description, etc.

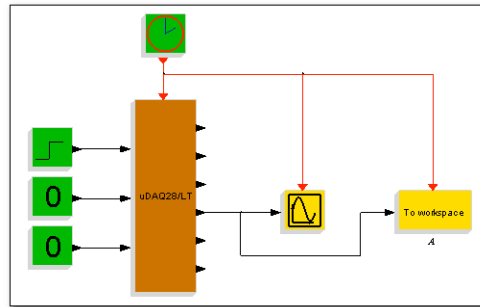


Figure 3. Measurement of transient response using uDAQ28/LT block

D. Scicos Block for Communication with Web Application

Another task that has to be solved is the question how to communicate with Scicos from the web page. Since it is supposed that experiments will not run on the web server where web pages are situated, it was necessary to solve the transfer of data via Internet. Standard installation of SciLab/Scicos doesn't contain any functions or blocks for a network communication. Therefore we decided to create a client-server application communicating via TCP/IP sockets. The client part on the Scicos side is realized by the new created autonomous Scicos block that is called SockCom. This block sends data from simulation to the server and operates client requests for closing the simulation. Similarly as in the previous case this block was also built in two steps: firstly, it was necessary to develop its executive part and then it was necessary to prepare its visualization. This block has only input ports since it always operates as the output block. The block has 3 parameters where we can define size of input signals, server IP address and port that will serve for connection during the simulation (Fig.4).

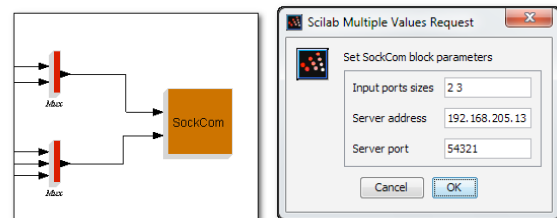


Figure 4. SockCom block and its parameters

E. Scicos Block Scheme

The Scicos block scheme for the remote control of the thermo-optical plant is shown in Fig. 5. Except of two blocks that were presented in previous paragraphs it contains a block for a controller. We considered two types of controller blocks: PID controller and Scifunc block that enables user to define own algorithm for the control of the plant. Whereas the PID block allows user to set up only PID controller parameters, the Scifunc block is more complex. However, in this case it is necessary to take into account the fact that a remote user can specify in the block any code that is executable in the SciLab script language, i.e. not only a control algorithm. This code can be eventually dangerous, because SciLab also enables to write any text to a file and subsequently to execute this file by the operating system. Therefore it is necessary to

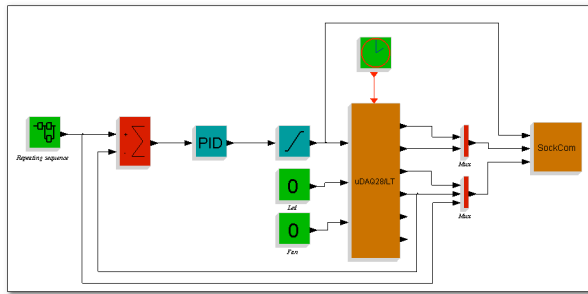


Figure 5. Scicos block scheme for the remote control of the thermo-optical plant (uDAQ28/LT block)

realize the whole solution only on the secured server whereby the security has to be solved on the level of operating system. The web server can run under the restricted user that has only minimal permissions necessary for interaction with SciLab and our client/server application.

F. Socket Server

The task of this server is to process requests from the end user, to ensure the control of the real device and to send experiment results back to the web page. The server is developed in Python and it is a simple console program without a graphical interface that has to run on the computer where the thermo-optical plant is connected to and where SciLab is installed.

The communication with the server is accomplished in an asynchronous way. It means that the server can deal with several requests in the same time. It is very convenient since this application has to communicate both with SciLab (using SockCom block) and the script (called SockPHP) that dynamically generates the web page for the end user (Fig.6).



Figure 6. Socket Server communication

G. Web Portal

The web portal (Fig.7) enables the interaction of the user with the remote experiment. It has several parts

- theoretical description of the experiment,
- time reservation (in each time the experiment can be run only by one user),
- setting parameters, i.e. parameters for simulation and controller; user is also allowed to set up own control algorithm,
- result visualization.

The administrator is also allowed to see the time reservation of all other users. Except of that the administrator has the right to create a new experiment that can be immediately visualized for students.

VI. CONCLUSIONS

One way how to increase the interest of students in technical education is to make the educational process more attractive. Except of the theory students need to master various practical problems. Numerical calculations

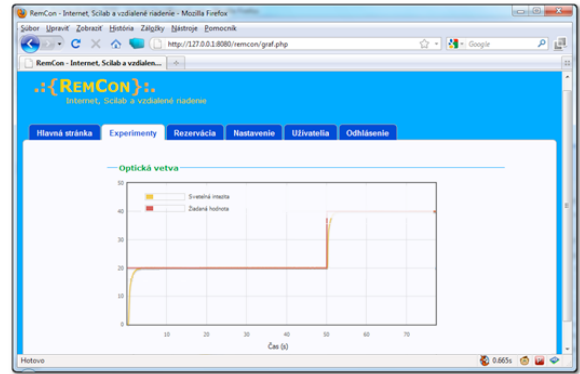


Figure 7. Web Portal for remote control of thermo-optical plant

can be supplemented by interactive exercises, tasks, simulations and real experiments that result from real situations and not only from the abstract concept. Building of remote laboratories helps this direction. Moreover, they are available to students when they need them and not only when the laboratory room is opened.

The presented approach to remote control uses the engine of SciLab/Scicos simulation software package. The SciLab is usually used for computations that are accomplished on the computer locally. However, the increased expansion of Internet together with the growing support of online education raised a question how to exploit capabilities of SciLab for these purposes, too.

The SciLab environment doesn't bring any standard solution how to exploit this software over Internet. However, there exist some indirect ways. One of them was introduced in this paper.

ACKNOWLEDGMENT

Authors thank to Zoltán Magyar for his help and discussions.

REFERENCES

- [1] G P. Bisták, "Remote Control of Thermal Plant Using Easy Java Simulation", *Int. Conf. on Interactive Computer Aided Learning ICL'06*, Villach, Austria, 2006.
- [2] R. Bucher, "Scicos RTAI", <http://www.scicos.org/scicosrtai.html>
- [3] J. Erdfelt, D. Drake, "LibUSB Homepage", <http://www.libusb.org/>
- [4] M. Huba, "Thermo-Optical Laboratory Plant uDAQ28/LT", <http://www.eas.sk/mod/product/show.php?ID=5>, 2008.
- [5] M. Huba, P. Bisták, M. Fikar, M. Kamenský, "Blended Learning Course 'Constrained PID Control' ", *7th IFAC Symposium on Advances in Control Education ACE'06*, Madrid, Spain, 2006.
- [6] N. Kroes, "Being open about standards", *OpenForum Europe - Breakfast seminar*, Brussels, 10th June 2008, <http://europa.eu/rapid/pressReleasesAction.do?reference=SPEECH/08/317>
- [7] S. Manori, R. Nikoukhah, S. Steer, "Scicos HIL Hardware In the Loop", <http://www.scicos.org/scicoshil.html>, 2006.
- [8] S. Mottelet, A. Pauss, "XMLlab Homepage", <http://marguerite.lmac.utc.fr/>.
- [9] M. T. Restivo, J. Mendes, A.M. Lopes, C.M. Silva, F. Chouzal, A Remote Lab in Engineering Measurement, *IEEE Trans. on Industrial Electronics*, vol. 56, no.12, pp. 4436-4843, 2009. <http://dx.doi.org/10.1109/TIE.2008.2011479>
- [10] F. Schauer, M. Ožvoldová, F. Lustig, "Real Remote Physics Experiments across Internet – Inherent Part of Integrated E-Learning", *Int. Journal of Online Engineering (iJOE)*, 4, No 2, 2008.
- [11] Chr. Schmid, "Internet - basiertes Lernen", *Automatisierungstechnik*, 51, No. 11, p. 485-493, 2003. <http://dx.doi.org/10.1524/auto.51.11.485.19585>

PAPER
SCI/LAB BASED REMOTE CONTROL OF THERMO-OPTICAL PLANT

- [12] M. Šimunek, P. Bisták, M. Huba, "Virtual Laboratory for Control of Real Systems", *Conference Proceedings ICETA*, Košice, Slovakia, 2005.
- [13] I. Zolotová, M. Bakoš, L. Landryová, "Possibilities of communication in information and control systems", *Annals of the university Craiova, Series: Automation, Computers, Electronic and Mechatronic*, Vol.4(31), No.2, pp.163-168, ISSN 1841-062, 2007.
- [14] K. Žáková, M. Sedlák, "Remote Control of Experiments via Matlab", *Int. Journal of Online Engineering (iJOE)*, 2, No. 3, 2006.

AUTHORS

M. Jáno, K. Žáková are with the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: miroslav.jano@gmail.com, katarina.zakova@stuba.sk).

The work has been partially supported by the Slovak Grant Agency, Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09. This support is very gratefully acknowledged. This article is a modified version of a paper presented at the International Conference REV2011, held at Transylvania University of Brasov, 29 June – 1 July 2011. Received, 5 August, 2011. Published as resubmitted by the authors 26 October 2011.