

Matlab Based Remote Control of Thermo-Optical Plant

[doi:10.3991/ijoe.v5s2.1095](https://doi.org/10.3991/ijoe.v5s2.1095)

K. Žáková and M. Kohút

Slovak University of Technology, Bratislava, Slovakia

Abstract—The paper is focused on a thermo-optical plant that is used for experimental work at the Faculty of Electrical Engineering and Information Technology STU in Bratislava. Our attention was dedicated to one approach of its remote control. A Java client-server application was created that enables to connect the graphical user interface with the Matlab engine running on the server.

Index Terms—client application, Matlab, remote control, server application, thermo-optical plant.

I. INTRODUCTION

Experiments play a very important role in the process of education. They enable to understand better the topic they are related to. In addition to theory students need to master various practical problems. Numerical calculations can be supplemented by interactive exercises, tasks, simulations and real experiments that result from real situations. The main motivation for using plants in the educational process is the clear physical “visibility” of the controlled dynamics, and also the necessity to exercise all design steps starting with the plant identification and ending with the evaluation of control results achieved with the particular model. The presented plant enables to exercise all these steps of the control design.

However, the question arises how to make an experiment available to all students at the time when they need it. The development of virtual and remote laboratories [2, 4, 6, 8, 9, 10, 11, 12] helps in this direction. In addition, this type of experimentation facilitates resource sharing among various institutions and improves national and international cooperation. The difference between virtual and remote laboratories consists in the fact that while in a virtual laboratory all the experiments are only simulated, in a remote laboratory students are able to control a real experiment from a remote computer.

Sometimes it can seem that remote experiments are similar to simulation techniques because they are easy to run and they require minimal space and time. These experiments can usually be rapidly configured. However, in contrast to simulations, they provide real data. The paper demonstrates one approach for establishing the remote control of a thermo-optical plant. The solution has already been tested and the experiment works on-line very well.

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 [3] has three inputs – the bulb voltage (the heat & light source), the ventilator voltage (system cooling) and a 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. 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 wide-spread.

The communication with the computer runs via the string exchange. The data transfer rate is 250kbit/s. The plant is shipped with the driver for Matlab/Simulink environment and therefore control algorithms can be easily set via this programming tool. The next benefit consists in the fact that thanks to the plant construction the Matlab simulation scheme does not need to be compiled. It enables that the control algorithm can also be written using a common “Matlab Fcn” block that can be found inside Simulink. 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.



Figure 1. The combined thermo-optical laboratory plant

As it is possible to see the introduced plant offers a big variability of experiments that can be accomplished. In addition to control students also have to solve tasks that are connected with plant identification, input-output data manipulation and communication with outer computer environment.

III. REALIZATION

The basic problem that has to be solved when building an online experiment is to find a way for communication between remote client and 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.

A. Server Side

The server part can be solved in several ways. One approach comes from the fact that interaction with Matlab is accomplished via built-in API libraries that can be exploited by means of the Java programming language [5]. This object oriented language can be easily integrated into the web server. In addition, it supports multi-thread communication that enables to run several Matlab applications. Stefan Müller introduced JMatLink library [7] that can be used for remote control of Matlab applications. JMatLink enables to use MATLAB's computational engine inside our own Java applications, applets or servlets. This dll library is programmed in C language that using ActiveX tool approaches data and services of Matlab (Fig. 3). The library is implemented in the class of Java language that has the same name (JMatLink.class).

In the case that Java is used for development of an application one has available all methods of the mentioned class. JMatLink uses a multi-threading approach to improve performance and handle multiple Matlab sessions at a time. As it was already mentioned the JMatLink library enables the connection to Matlab from Java servlets. These so called JMatServlets can be later approached from the web browser via HTTP. The servlet functionality can open and initialize the Matlab engine, start all simulations, get results and send them to the client application. Of course, it finally enables to close the connection with Matlab application.



Figure 2. Communication between remote client and technological plant.

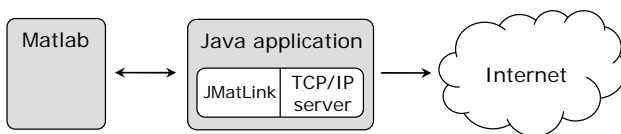


Figure 3. Communication between Java application and Matlab via JMATLink library

The created Java based server application was tested on a PC with Matlab v.6.5 and Matlab v.7.0. The server that takes care of communication with Matlab has to be supplemented with the web server where the client application is placed. We decided to use Apache Tomcat 6.0 server with the support of PHP programming language.

B. Client Side

The client application is dynamically generated by the web server. It serves for the user interaction with the remote system. In addition to PHP we also used the AJAX approach that enables asynchronous modification of the web page.

For the client side we developed an application that enables to put virtual and remote experiments on the web (Fig. 4 – button “Add”).

The created environment can run in 2 modes. The first one is determined by the experiment administrator. Here one can set a new experiment, enter experiment description, upload related Matlab files and activate the prepared experiment for online running.

The second mode is devoted to the experiment itself. It can be realized as a virtual simulation or in the form of remote participation in the laboratory. In this paper our attention is devoted to the remote realization.

The remote control of the plant consists in the initialization of the connection with Matlab, initializing and starting the simulation, getting results and closing the session. The user can see results using the dynamic graph and after simulation he or she can download experimental results to the local file.

The graphical user interface for the experiment control is shown in Fig. 5. In the illustrated example we controlled temperature in the plastic cylinder with a classic PID controller. The student receives a predefined set of controller parameters and his or her task is to improve result performance. In addition, the interested user is offered the possibility to change the default control algorithm. The modification of the control structure can be accomplished by the change of the Simulink block scheme. The remote user can upload 2 files to the server: one for the modified block scheme with predefined inputs and outputs (mdl file) and one for the numerical values of parameters (mat file) (Fig. 6).

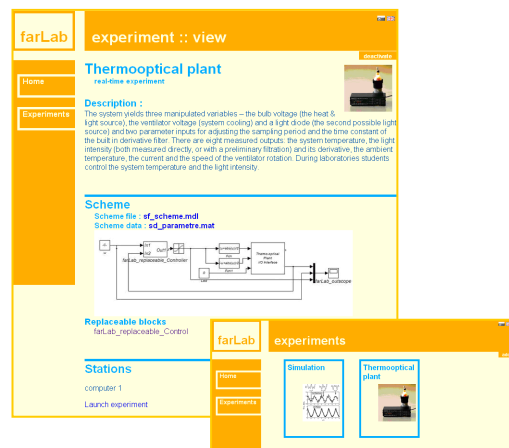


Figure 4. Plant presentation and list of available experiments



Figure 5. Realization of remote experiment

These files are completely designed by the remote user; he or she has to follow only very simple rules to keep the parameter notations. To prevent possible harm from the remote users we decided that they will be forbidden to use “Matlab Function” and “S-function” blocks in the Simulink block scheme. Remote users are allowed to use only blocks where it is not possible to run any command that might execute some external program.

For interactive visualization of results it was necessary to choose a suitable graphical data format. It should be standard and supported by web browsers. We decided to use the very promising SVG format because it is a vector format that thanks to its XML structure can be easily modified, zoomed, animated, etc.

IV. CONCLUSIONS

Many approaches to the remote control use the engine of the very popular Matlab simulation software package. It facilitates the whole design procedure because it is only necessary to take care of the communication via Internet and not of computations and tasks that are accomplished by Matlab, e.g. the numerical methods needed for simulations.

One way to realize the Matlab based remote experiment was demonstrated in this paper. The connection of Matlab with Java application on the server side enabled to build a stable remote experiment. The solution with Java was also presented e.g. in [1]. The advantage of the present approach is that it enables to build a very friendly graphical user interface where the remote user can also set control algorithms designed by him.

It should be noted that the main attention was dedicated to the possible technology. Nevertheless building the remote experiment still requires solving some complementary tasks such as the experiment administration or its safety issues. The administration should ensure that in each moment only one user can approach the real device and that everybody will have time to accomplish their own experimental work. Therefore it is very important to protect the equipment from possible damage and to protect people close to the place where the experiment runs. Solving these questions should not be omitted.

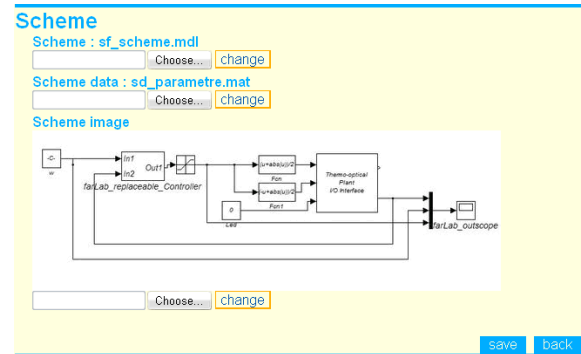


Figure 6. Modification of control scheme

REFERENCES

- [1] P. Bisták, “Remote Control of Thermal Plant Using Easy Java Simulation,” in *Proc. Int. Conf. on Interactive Computer Aided Learning ICL'06*, Villach, Austria, 2006.
- [2] P. Bisták, K. Žáková, “Organising Tele-Experiments for Control Education,” in *Proc. 11th Mediterranean Conference on Control and Automation*, Rhodes, Greece, 2003.
- [3] M. Huba, “Thermo-Optical Laboratory Plant uDAQ28/LT,” <http://www.eas.sk/mod/product/show.php?ID=5>, 2008.
- [4] F. Jakab, V. Andoga, L. Kapova, M. Nagy, “Virtual Laboratory: Component Based Architecture Implementation Experience,” *Electronic computer and informatics*, Košice-Herľany, Slovakia, September, 2006.
- [5] Klimke, “How to Access Matlab from Java,” *Berichte aus dem Institut für Angewandte Analysis und Numerische Simulation*, Universität Stuttgart, Preprint 2003/005, 2003.
- [6] J. Liguš, J. Ligušová, I. Zolotová, “Distributed Remote Laboratories in Automation Education,” in *Proc. 16th EAEEIE Annual Conf. on Innovation in Education for Electr. and Information Eng.*, June, Lappeenranta, Finland, 2005.
- [7] S. Müller, “JMatLink Version 1.00,” <http://www.held-mueller.de/JMatLink>.
- [8] 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.
- [9] Chr. Schmid, “Internet - basiertes Lernen“, *Automatisierungstechnik*, 51, No. 11, pp. 485-493, 2003.
- [10] M. Šimuněk, P. Bisták, M. Huba, “Virtual Laboratory for Control of Real Systems,” *Conference Proc. ICETA*, Košice, Slovakia, 2005.
- [11] K. Žáková, M. Huba, V. Zemánek, M. Kabát, “Experiments in Control Education,” in *Proc. IFAC Symposium on Advances in Control Education*, Gold Coast, Australia, 2000.
- [12] K. Žáková, M. Sedlák, “Remote Control of Experiments via Matlab”, *Int. Journal of Online Engineering (iJOE)*, 2, No. 3, 2006.

AUTHORS

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

This work was supported in part by the Slovak Grant Agency, Grant KEGA No. 3/7245/09.

This article was modified from a presentation at the IRF'2009 conference in Porto, Portugal, July 2009. Submitted 09 October 2009. Published as resubmitted by the authors on 20 October 2009.