

# Remote Control of Experiments via Matlab

K. Žáková and M. Sedlák

Faculty of Electrical Engineering and Information Technology  
Slovak University of Technology, Bratislava, Slovakia

**Abstract**—The paper demonstrates two possibilities of remote control of real experiments using Matlab software environment. We introduced basic approaches to Matlab remote access using Matlab Web Server, COM and Virtual Reality Toolbox. The proposed concepts were used and tested for control of magnetic levitation system. The gained experience was also discussed in the paper.

**Index Terms**—Internetworking, Remote handling, Virtual reality.

## I. INTRODUCTION

The popularity of online education is permanently growing. It can be denoted also as virtual education, Internet based education, web based education, and education via computer-mediated communication. In engineering education providing software and hardware components of existing laboratories via the Internet creates a base for establishment of virtual and remote laboratories. The main motivation for using plants in the educational process is 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 the control results achieved with the particular model.

Design of control applications that are available via Internet is oriented in two directions: control of virtual devices and control of real physical plants. The first possibility enables to simulate the virtual model on a computer and with Internet access to offer it as an animation to students via the WWW or CD-ROM. However, using the animation models cannot substitute the work with real physical plants that always demonstrate some unmodelled dynamics, parasitic noise, friction, etc. Unfortunately, the number of students is high in comparison with the number of available real plants. A possible solution of this problem is building of remote lab that gives learners access to laboratories via Internet.

## II. PROBLEM DESCRIPTION

### A. Real plant

For the illustration of the problem we decided to use the magnetic levitation model that is provided by Humusoft company [7]. The model of magnetic levitation demonstrates control problems associated with nonlinear unstable systems. The system consists of a coil levitating a steel ball in magnetic field. Position of the steel ball is sensed by an inductive linear position sensor connected to A/D converter. The coil is driven by a power amplifier connected to D/A converter. The basic control task is to control the position of the ball freely levitating in the magnetic field of the coil. The magnetic levitation system is a nonlinear dynamic system with one input and one output. The system is designed to be controlled by digital

controllers, but as of its nature using analog controllers is also possible.

### B. Requirements

Our aim was to control the introduced plant remotely. We had several requirements for the application. We decided to use Matlab Real-Time Workshop for real-time simulation of the plant. We wanted to have partial graphical results during the whole run of simulation and not only after the simulation ends. For better illustration of the experiment we took a decision to visualize it by means of 3D animation using VRML model that is directly incorporated in Matlab.

## III. REALISATION

We investigated two alternative methods that can be used for remote control of plants and that satisfy the requirements specified above. They include Matlab Web Server and Component Object Model.

### A. Matlab Web Server

The Matlab Web Server enables to create Matlab applications and to place them on the web. Programmers can easily use the capabilities of the World Wide Web to send data to Matlab for computation and to display results in a Web browser. Users of the application do not need to know Matlab environment, they only fill the prepared form with suitable parameters and after sending it for processing, they receive back the expected response.

From the technical point of view after editing the parameters is completed, the client sends data to the HTTP daemon, which loads Matlab web server (matweb file) through Common Gateway Interface (CGI). Matlab web server connects to the Matlab server by means of internal protocol. Matlab server loads the requested M-file (stored locally in the directories of the server) into a separated copy of Matlab. The file is executed that in our case means that Matlab starts simulation and it approaches the real plant via A/D card using Real Time Workshop.

The standard procedure is designed in such a way that after simulation executing has been completed the Matlab server gives an output to the Matlab web server, usually as text/html based on a template file. Then, the Matlab web server returns it to the client through HTTP daemon. However, in such a way, the user receives all results after the simulation process. Our requirement was to have partial results also during the simulation. Therefore, the information about the position of the ball in magnetic levitation model is collected by the proposed S-function in the Matlab/Simulink model. In regular intervals the S-function sends command to Matlab API to generate a figure with the simulation results. Matlab creates the required figure according to the predefined parameters specifying the path, name, format, resolution and size of the final file. After it sends the command to the server

operating system to save the file to the local hard disk. PHP script enables the visualization of the figure to user. Since the web page in the user web browser is automatically and regularly refreshed, the results are permanently actualized.

In Fig.1 it is possible to see the proposed structure of the remote control. Matlab server manages the communication between the Web application and Matlab. The Matlab server presents a multithreaded TCP/IP server. Matlab web server is a TCP/IP client of Matlab server. This program uses the Common Gateway Interface (CGI) to extract data from HTML documents and transfer it to Matlab server [8].

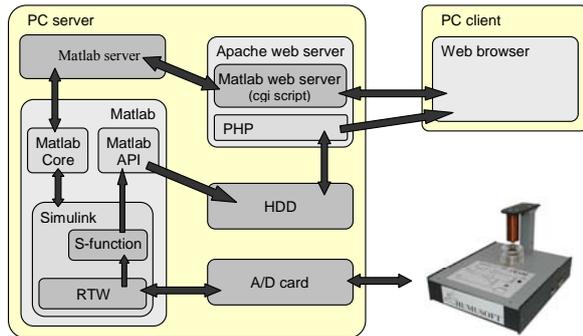


Figure 1. Remote control via Matlab web server

**B. Component Object Model**

Component Object Model (COM) is a software architecture developed by Microsoft to build component based applications that can be called up and executed in a Windows environment. A program can call the object whenever it needs its services. Standard applications, such as word processor and spreadsheets [9], can be written to expose their internal functions as COM objects, allowing them to be “automated” instead of manually selected from a menu. For example, a small script could be written to extract data from a database, put it into a spreadsheet, summarize and chart it, all without manual intervention. The similar functionality is also supported by Matlab. The user can approach the most of object parameters via interface that enables communication between COM client and COM server. Matlab supports the only interface and it is called IDespatch. Except of the interface the programmer needs to know what approach methods he or she can use. The simplest one is the already mentioned Automation method that is also supported by Matlab.

The proposed method is sketched in Fig.2. The whole application is initialized by the user that from the web browser runs PHP script placed on the web server. PHP creates a COM client and sends it one parameter that denotes the application name to open. Using server operating system the COM client executes Matlab in the Automation mode, i.e. it creates COM server object that waits for next instructions. COM client addresses the COM server and sends there simulation parameters and commands for building the block scheme, connecting it to the real system and starting the simulation. Then, the COM server starts the simulation and sends control data using Real Time Workshop via A/D card towards the real plant (magnetic levitation model). The next steps are the same as in the previous case. The data are collected by S-

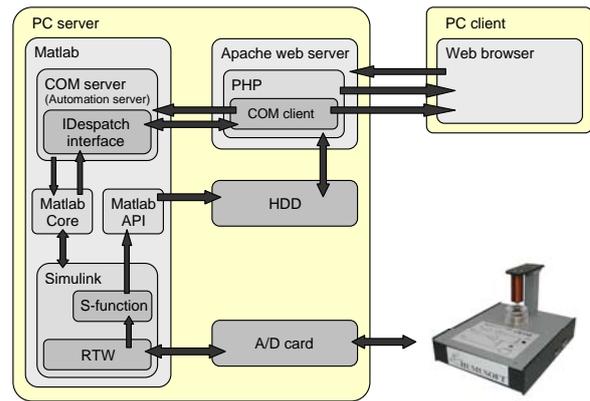


Figure 2. Remote control via COM

function that takes care for the results processing. The results are again saved on the local hard disk in the form of pictures that are later displayed in the user web browser.

It is to note that Matlab application cannot be opened before the COM object is created. Matlab always has to be initialized by created COM client because later they cannot be assigned each to other.

**C. 3D Visualisation**

Both previously mentioned methods enables to run simulation on the remote real plant model and to give continuous information about the output variable of the system. In the case of magnetic levitation system it is the position of the ball in the magnetic field. However, for some users this information can be very abstract and therefore it is convenient to supplement it by the suitable animation.

The Matlab Virtual Reality Toolbox is the server side package that is used for animating and visualizing systems in three dimensions.

It ensures data exchange between Matlab and the virtual scene (Fig.3). The toolbox includes the virtual reality server (VR server) that is actually a HTTP server operating client requests. The client connects to the server via standard Web browser that has to contain plug-in for visualization of VRML models and Java support. It can happen that it is necessary to accomplish the additional installation of these components. After the connection VR server checks status of all available VRML models and it returns their links to the client. The user can choose one of them. The click on the link opens the VRML viewer that loads Java communication interface and VRML plug-in connects to the server.

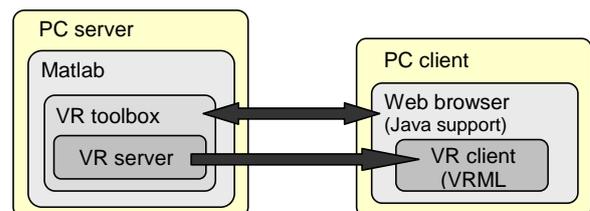


Figure 3. Remote control via COM

IV. WEB APPLICATION

In spite of the fact that we have verified two different approaches for the remote control of the magnetic levitation system, user can see only one graphical design of the web application for both of them. Its front side is shown in Fig.4.

The whole window is divided into four parts. On the left side it is possible to set all parameters for simulation. The user can select the input signal of the control scheme among step signal, generator of rectangular pulses and the input from VRML environment. Then, he or she can set parameters of PID controller that are used for the plant control. The last text item serves for the definition of simulation time. The user can also determine whether the 3D animation of the controlled plant should be displayed or not. In the positive case it is opened in the new window. Finally, the button "START" initiates the simulation.

The central part of the window serves for graphical visualization of simulation results that can be switched between two views: the first one shows actual position of the ball and the second one enables to display previous results that are saved in png files listed on the right side of the window. In these files there is saved either output or control variable. The user can distinguish the variable in the figure according to the file name – the figure with output starts with the letter "y" and the control variable with the letter "u". The name of the file also specifies the time instant in which the file was saved.

The bottom part of the window serves for the listing of commands that are sent to Matlab for their processing.

V. CONCLUSIONS

Our attention was dedicated to the development of techniques that enable not only to accomplish simulation in remote laboratory but also to offer to user the actual information about its running. For this purpose we tried to exploit Matlab since, thanks to its toolboxes, it facilitates access to the real plant experiments and it belongs to widespread applications in the control education. In this paper we described only two possible approaches to the remote control via Matlab. The Matlab Web Server introduces a standard solution that brings to programmer a lot of advantages from the design of web page layout to the guaranty of the application stability. The only disadvantage can be financial aspect since it has to be bought as an additional component of Matlab. The COM approach brings an alternative solution that can ensure the similar functionality as the Matlab Web Server. Other not mentioned approaches include the use of e.g. Matlab Dynamic Data Exchange (DDE), TCP/IP communication via S-function, etc.

However, in this moment it is necessary to say that introduction of all kinds of laboratories reasonably increases the student motivation and also develops various other skills in the signal measurement and processing. Remote laboratories enable to increase the quality of

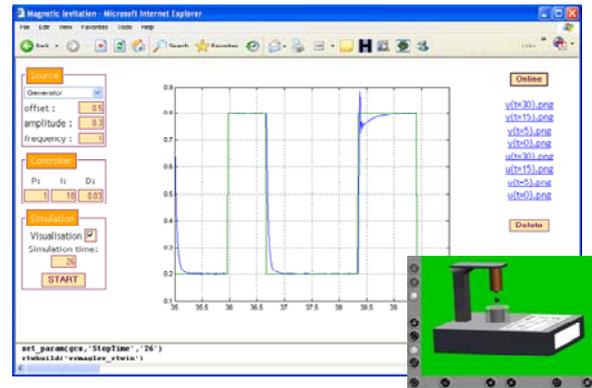


Figure 4. GUI for remote control of magnetic levitation

distance learning in engineering education to the level comparable with the face-to-face study.

ACKNOWLEDGMENT

The work has been supported by the Slovak Grant Agency, Grant No. VEGA 1/3089/06 and the Grant KEGA 3/3121/05. This support is very gratefully acknowledged.

REFERENCES

- [1] P. Bisták, K. Žáková, "Organising Tele-Experiments for Control Education," *11th Mediterranean Conference on Control and Automation*, Rhodes, Greece, June 2003.
- [2] F. Jakab, V. Andoga, L. Kapova, M. Nagy, "Virtual Laboratory: Component Based Architecture Implementation Experience," *Electronic computer and informatics*, Košice-Herľany, Slovakia September 2006.
- [3] J. Liguš, J. Ligušová, I. Zolotová, "Distributed Remote Laboratories in Automation Education," *16th EAEEIE Annual Conference on Innovation in Education for Electrical and Information Engineering*, Lappeenranta, Finland, June 2005.
- [4] P. Piš, K. Žáková, "Remote Control of the Beam and Ball Model," *Process Control 2005*, Štrbské Pleso, Slovakia, June 2005, pp.201.1–201.6.
- [5] Chr. Schmid, "Virtual Laboratory for Engineering Education," *ICDE conf.*, Vienna, Austria, 1999.
- [6] K. Žáková, M. Huba, V. Zemánek, M. Kabát, "Experiments in Control Education," *IFAC Symposium on Advances in Control Education*, Gold Coast, Australia, December 2000.
- [7] <http://www.humusoft.cz/models/ce152.htm>
- [8] <http://www.mathworks.com>
- [9] <http://www.answers.com>

AUTHORS

**K. Žáková, M. Sedlák** 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](mailto:katarina.zakova@stuba.sk)).

Manuscript received 18 July 2006.