

# MATLAB Web Server and M-file Application

Suzana Uran and Karel Jezernik

University of Maribor/Faculty of Electrical Engineering and Computer Science, Maribor, Slovenia

**Abstract**—Using of MATLAB Web Server(MWS) is attractive for students as well as for universities due to reduction of costs for both parties. Standard MWS applications support structured hands-on experiments welcomed for visualization of hard to grasp concepts and as a design tool. MWS applications for Math, Power electronics and Control design, part of a virtual laboratory created at University of Maribor, are represented. In addition a M-file MWS application - an example of unstructured MWS environment allowing students to write the MATLAB M-files of their own and to execute them on MWS is presented in detail. Only a simple text editor and an Internet browser are needed by a student to use M-file MWS application.

## I. INTRODUCTION

In engineering education, MATLAB is one of the software tools used by over 2500 Universities all over the world to support learning. It is also used at the Faculty of Electrical Engineering and Computer Science, University of Maribor. Over the last five years, a hands-on introduction to the MATLAB has been given to our undergraduate mechatronics and automation students. Lack of time and lab availability motivated us to develop MATLAB Web server applications. MATLAB Web Server(MWS) enables access to MATLAB and MATLAB based learning applications at any time and from any place (school or home). In this way student and university costs are reduced at the same time while a greater availability of MATLAB for students is achieved.

Among the teaching community a great variation of needs exists regarding the using of MATLAB. These variations stem from teacher's goals, students background knowledge and skills, the environment where the learning should take place and costs. Therefore some teachers welcome applications for hands-on experiments where no programming or math is needed to perform the experiment [1], [2], [3], [4]. Such hands-on experiments are fully structured and don't allow students to create experiments of their own. On the other side engineering education [5] must assure that young engineers possess the ability to design and conduct experiments, as well as to analyze and interpret data, and the ability to design a system, component, or process to meet desired needs. It is also not very likely that the problem students would have to solve in their professional carrier would fit into one of the structured exercises they learned at school. Taking into account these facts there is a permanent need in engineering education for less structured environments where students could build experiments of their own and practice design and data analyzes and interpretation.

Using of MWS is very attractive for students as well as for the university due to cost reduction as stated in the first paragraph. In our efforts to introduce MWS into our

curriculum we found that MWS is well suited for fully structured hands-on experiments over the Web. Unfortunately, MWS documentation shows no way how to use MWS in less structured way [6], [7] - the way MATLAB is programmed (used) with M-script locally. In order to overcome this drawback we wrote M-file application for MWS. M-file application allows our students writing of their own M-files and sending them to MWS for execution. The results are obtained over the Web. To use M-file application students need only an Internet browser and any of the text file editors, for instance Notepad. Therefore student's costs are not increased since Notepad or similar text editors are usually part of the computer operating system. The problem of using MWS was also solved in [6]. A comparison of both solutions is represented in section III.

In section II MWS is briefly described and our standard MATLAB Web server applications for three different courses are presented. In Section III our M-file application for MWS is represented in detail. This application enables students to write their own M-code for MATLAB and design their own MATLAB experiments. At the end of the section III our solution is compared with the solution presented in [6].

## II. MATLAB WEB SERVER AND APPLICATIONS

The structure of the MATLAB Web Server(MWS) and it's dependency on computer platforms is considered in [7]. Our MWS is configured on Apache Server. Currently we use MATLAB 6.5 release 13 and accompanying MWS.

The MATLAB Web Server(MWS) enables us to create MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results in a Web browser. The MWS depends upon TCP/IP networking for transmission of data between the client system and MATLAB. We use the simplest configuration where a Web browser runs on a client Personal computer (PC) while the MATLAB Web Server and the Web server daemon run on the server machine.

MATLAB Web Server applications are a combination of M-files, HTML documents and graphics. Two HTML documents are needed. The first HTML document collects the input data from the user, sends them to the MWS and runs a M-file needed for the application. The M-file needed for the application resides permanently on MWS, in the following we will refer to it as an application M-file. The application M-file reads input data collected by the first HTML document from a MATLAB structure, performs requested operations, generate requested graphics and places the output data into the MATLAB output structure. The second HTML documents gets results from the MATLAB output structure and displays

them on the client machine. For detailed information see MATLAB Web Server Documentation.

In the following we will refer to the previously described applications as the standard MATLAB Web Server applications as opposed to M-file MATLAB Web Server application (also M-file application for MWS).

A virtual laboratory based on standard MATLAB server applications was built for the needs of the educational process at the Faculty of Electrical Engineering and Computer Science. Internet address of our virtual laboratory is: <http://hl1.uni-mb.si>. In this section three different applications are represented.

### A. Matrix eigenvalue application

Matrix eigenvalue application is an application for Math. It was prepared for freshmen (1<sup>st</sup> year) students. In Math eigenvalues and characteristic polynomials of 2x2 matrices are derived by hand. Students rarely handle 3x3 matrices to derive matrix eigenvalues and characteristic polynomial. An MATLAB Web Server application offers students the possibility to calculate eigenvalues and characteristic polynomial for matrices up to dimension 6x6. Due to low experiences in programming and using MATLAB standard Web Server applications are a very useful tool for hands on experiments in Math for freshmen students. Fig.1 shows an example of matrix eigenvalue calculation for a case of 4x4 matrix.

Another example of Math application on our MATLAB server is the application calculating the solution of a system of linear algebraic equations up to order 6x6.

### B. Buck converter application

Buck converter application is an example of application for Power Electronics course. The idea of the application is to enable active learning by a hands-on experiment and to visualize some concepts in Power electronics course. An example of such concept is the influence of PWM on action and performance of Buck converter. Fig.2 shows the web page for input of a Buck converter circuit parameters and an example of Buck converter results for variation of inductance L. For this application a Simulink block scheme is run by the MWS [8].

### C. Web sisotool application

Web sisotool application is an example of application for learning of control design. The idea for the application comes from MATLAB's tool called sisotool. For Web sisotool application the order of the transfer functions F, C, and G is limited to order three in order to keep input data and results visible on one web page. It was agreed that currently third order of transfer functions is sufficient for introduction to control. Two graphs – results are obtained at the same time. On the graphs between time response, rootlocus or bode plot could be selected. Interactivity of Web sisotool application is constrained in comparison to MATLAB's locally run sisotool to variation of transfer function parameters. It is an easy to use control design tool.

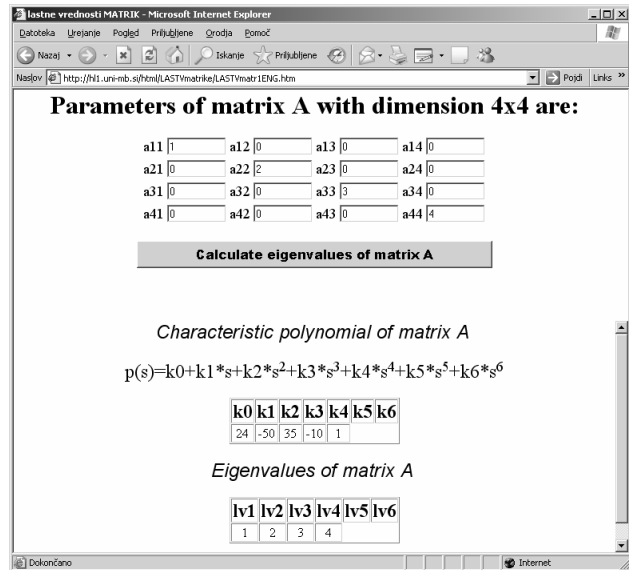


Fig.1: Matrix eigenvalue calculation for a case of 4x4 matrix

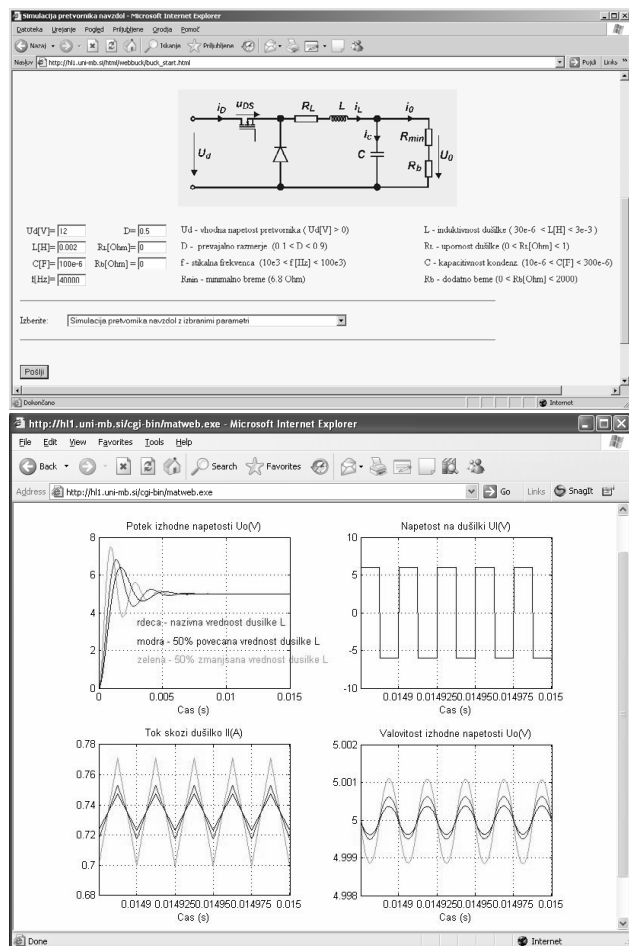


Fig.2: Buck converter application with results

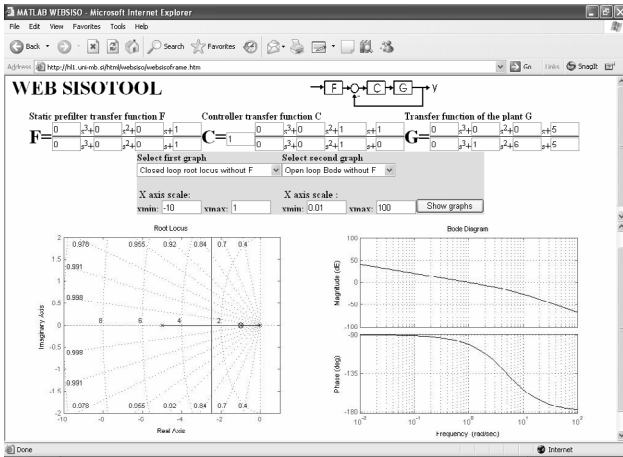


Fig.3: Web sisotool application

### III. M-FILE MATLAB WEB SERVER APPLICATION

The aim of the M-file MWS application is to allow our students(user) writing of their own M-files and sending them to MWS for execution. Students(user) obtain the results over the Web. The concept of our solution is to build one standard MATLAB Web Server application that would serve for the upload and execution of user 's M-files. Our desire was to implement the solution that would use the same programming code for M-file MWS application as it is used for execution of M-file with MATLAB locally. It should be noted that all user data should be defined inside the user M-file. Further we accepted a constraint that the results are only graphical and are displayed as a Figure. One Figure per user M-file is allowed.

For the implementation of the M-file MWS application we chose PHP programming language and MySQL database. PHP language was chosen because it could be mixed with the HTML language.

Fig.4 shows a diagram of M-file MWS application implementation.

For the standard MWS application two HTML files are needed and the application M-file. The task of the first HTML file is to collect data from user, send data to the MWS and run the application M-file on MWS. In our solution we replaced the first HTML file with two PHP files. The task of the first PHP files is to upload necessary user M-files. It is possible to upload up to three user M-files to the MWS at the same time if needed. The second PHP file serves for the reservation of user directory on MWS, for saving of uploaded user M-files to the user directory, for sending a user directory name and a name of user M-file to be executed as application data and to run the application M-file on MWS.

On MWS there is a limited number (temporary 15) of user directories available. The reservation of user directories is based in general on the user TCP/IP address and on MySQL database. Only one user could access MySQL database at the same time therefore two users cannot reserve the same user directory. In MySQL database the information whether the user directory is reserved or not, TCP/IP address of the user who reserved the directory and the time of the last file upload to the MWS is stored. To active users the same user directory is assigned on the subsequent uploads of user M-files on

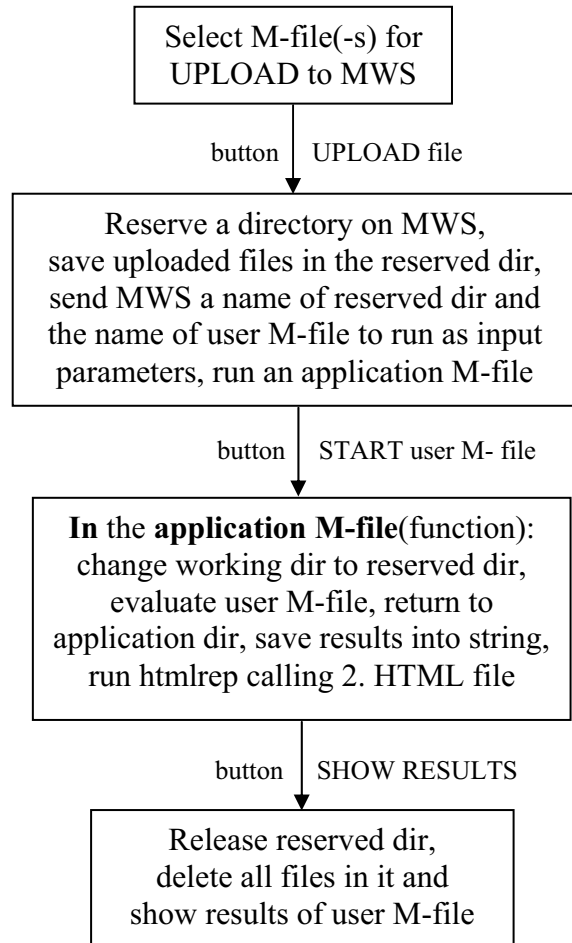


Fig.4: M-file application implementation for MWS

MWS. Reservation of the user directory of a passive user is canceled automatically if the subsequent user M-file upload does not follow in 15 minutes. If the user tries to reserve a directory when all user directories are reserved, the user is asked to try later. After the user has reserved the user directory all user's M-files are saved on it. And finally the execution of the application M-file is started and names of the user directory and of the user M-file to be executed are sent to it as application input data. In the case of two or three user M-files uploaded to MWS the user M-file to be executed is always the M-file uploaded with number 1. The remaining uploaded user M-files had to be called from the user M-file to be executed or they have no action.

The application M-file is a function according to the template from MWS documentation. In the application M-file MATLAB identification number and names of the application directory, the user directory and the user M-file to be executed are read from the input structure first. Then the directory is changed to the user directory and the user M-file to be executed is evaluated. To save results the directory is changed to the application directory and results are saved in a string named s. At the end the second HTML file is called using htmrep function and an output string s with results is sent to it.

In our solution the second HTML file notifies the user that the MATLAB has had finished the execution and calls the third PHP files. The third PHP file releases the

reserved user directory, deletes all user M-files and displays results.

Fig. 5,6,7,8,and 9 show the whole M-file application as is seen by the user on the browser when using MWS.

```

odziv - Notepad
File Edit Format View Help
% show step response
f=tf(1,[1 2]);
step(f)

```

Fig.5: M-file for step response called odziv



Fig.6: M-file upload to MATLAB Web Server

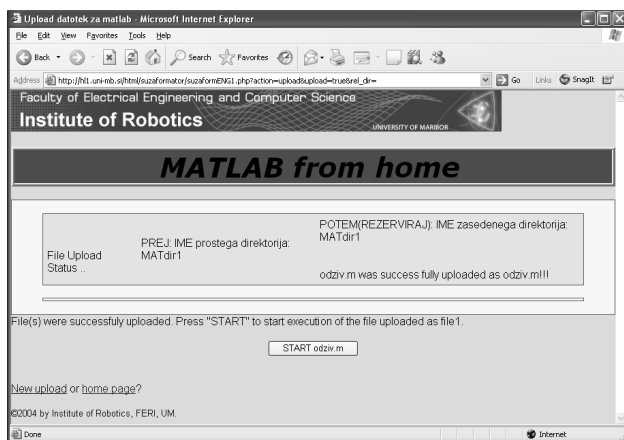


Fig.7: Start MATLAB Web Server execution of odziv

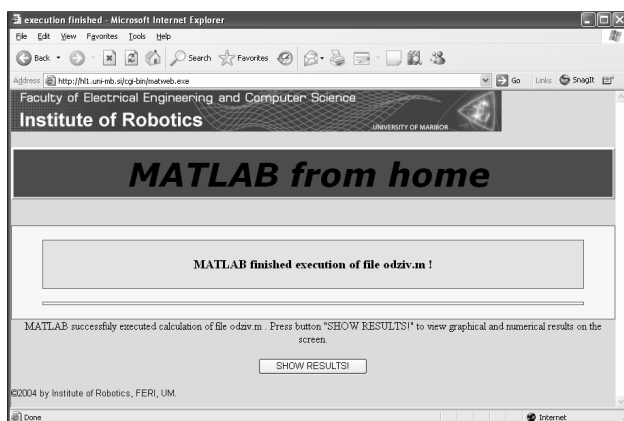


Fig.8: MATLAB Web Server execution finished

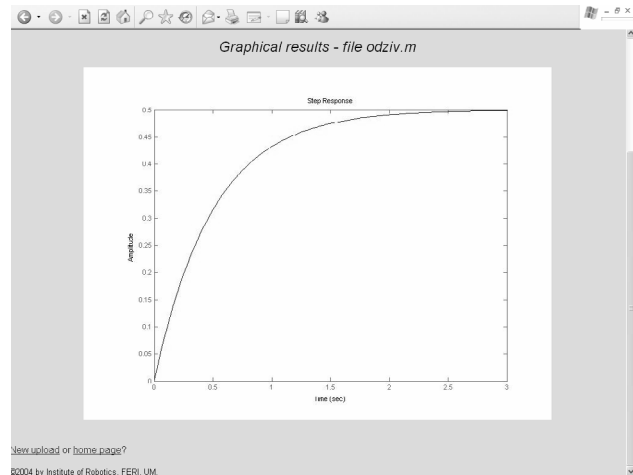


Fig.9: Step response of M-file odziv

In Fig. 5 a simple code named odziv for calculation of step response written in text editor Notepad is shown. In Fig.6 the web page for upload of M-files is presented. Fig. 7 and 8 show the web page for start of execution of M-file odziv and for final display of results. And in Fig.9 the resulting step response in form of a Figure is shown.

For M-file MWS application a use of Internet Explorer, as browser, is recommended but not necessary. A good side of our solution is that using Internet Explorer MATLAB M-code syntax errors are reported to the user. A problem for our solution represents interactive action of MATLAB environment. In user M-file to be executed by MWS every command must end with a semicolon! If this is not the case the browser error: "the page cannot be displayed" appears or a variable display along with second PHP file is displayed and M-file execution is terminated. Our solution temporary does not support numeric results for M-file MWS application. This is not a serious limitation since frequently for many experiments graphical results are desired.

In Fig.10 the strength of the M-file MWS application is shown. Multiple step responses are plotted in the same Figure for easier comparison of the gain influence on a closed loop.

To achieve the same goal as we in [6] a solution is presented called Extension of MWS. Extension of MWS automatically creates the standard MWS application, it is automatically creates configuration data, two HTML files and an application M-file, for each uploaded user M-file. Uploaded user M-files are stored in MySQL database. Input of user data is foreseen over the Web browser, as with standard MWS applications, and the output results could be graphical (one or more Figures) or numerical in the case of Extension of MWS. The user M-file could be uploaded as a text from the web or as a text file and it is preprocessed. The user input and output variables had to be an element of sinput and soutput structure, respectively, in order to be recognized as input/output variables by the automatic algorithm of the Extension of MWS. Unfortunately, in [6] is no report about what happens if the user M-file contains syntactic error(s) or if a semicolon at the end of a MATLAB command is missing. In [6] is either no report on how a group of uploaded user M-files is processed. Could one user M-file be a function?

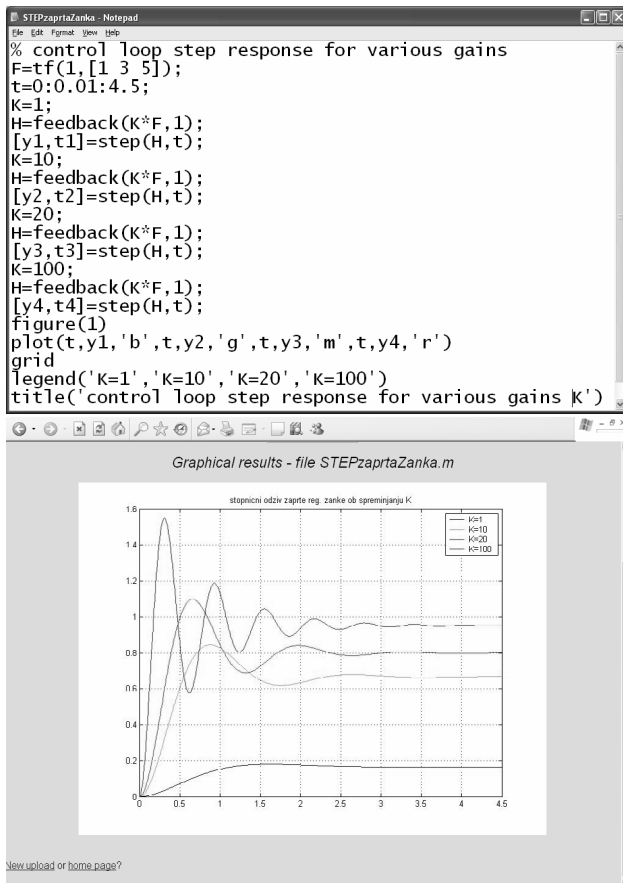


Fig.10: Multiple Step responses in one Figure

Extension of MWS is temporary a more general solution than ours because numerical results are possible in addition to graphical results. We are working towards adding a possibility of numerical results to our solution, too. The problem of the Extension of MWS is that for a transition between MATLAB used locally and MWS the user M-files must be changed by hand in order to put all input and output variables into sinput and soutput structure, respectively. The existing difference might be confusing for a student and an undesirable task for an experienced user. The difference does not appear with our solution. Another problem of the Extension of MWS is that considerable amount of code had to be generated for each uploaded user M-file what may result in memory troubles. In the case of our solution only one standard MSW application exist on MWS no matter what is the number of users.

#### IV. CONCLUSION

A virtual laboratory based on MWS was built at the Faculty of Electrical Engineering and Computer Science in Maribor. Applications for different courses: Math, Power electronics and Control, were built in order to encourage teachers to use it. Structured hands-on experiments were built along with an unstructured environment where students could write MATLAB M-files of their own and send them to MWS to be executed and thus create experiments of their own.. Only a simple text editor and Internet Browser are needed to perform experiments. Our virtual laboratory was successfully extensively used for teaching of Control since the fall 2005. M-file MWS application based experiments for Signals course and Simulation course were also successfully implemented. Power electronic course supported by hands-on experiment would start soon.

#### ACKNOWLEDGMENT

The authors would like to thank Martin Terbuc, Evgen Urlep and Alenka Hren for their help and effort in developing MATLAB Web Server applications.

#### REFERENCES

- [1] B. Wittenmark, H. Haglund, and M. Johansson, "Dynamic Pictures and Interactive Learning," *IEEE Control Systems*, vol. 18, pp. 26–32, June 1998.
- [2] B.L. Sturm and J.D. Gibson, "Signals and Systems Using MATLAB: An Integrated Suite of Applications for Exploring and Teaching Media Signal Processing," *Proceed. 35<sup>th</sup> ASEE/IEEE Frontiers in Education Conference*, pp. F2E-21–F2E-25, October 2005, Indianapolis, USA, Indiana .
- [3] M. de Magistris, "A MATLAB-Based Virtual Laboratory for Teaching Introductory Quasi-Stationary Electromagnetics," *IEEE Trans. on Educ.*, vol. 48, pp. 81–88, February 2005.
- [4] J. Sanches et al., "Easy Java Simulations: an Open-Source Tool to Develop Interactive Virtual Laboratories Using MATLAB/Simulink," *Int. J. Engineering Educ.*, vol. 21, No.5, pp. 798-813, September 2005
- [5] S.D. Creighton et al., "A Comprehensive System for Student and Program Assessment: Lessons Learned," *Int. J. Engineering Educ.*, vol. 17, No. 1, pp. 81-88, January 2001
- [6] M. Sysel and I. Pomykacz, "Extension of MATLAB Web Server," *Proceed. 35<sup>th</sup> IASTED Int. Conf. Advances in Computer Science and Technology*, pp. 70–74, November 2004, St.Thomas, US Virgin Islands.
- [7] A. Pester and R. Ismailov, "Interactive Applications in Teaching with the MATLAB Web Server ," *Vestnik National'nogo Techniceskogo Universiteta "KchPI"*, 22'2001, pp. 14-19, Kcharkiv 2001, Ukraine
- [8] A. Valera et al., "Virtual and Remote Control Laboratory Development," *IEEE Control Systems*, vol. 25, pp. 35–38, February 2005.