

# Remote system for development, implementation and testing of control algorithms

...

*Cvjetković Vladimir<sup>1</sup>, Dragoljub Stevanović<sup>2</sup>, Milan Matijević<sup>2</sup>*

Faculty of Science<sup>1</sup> Faculty of Mechanical Engineering<sup>2</sup>

**Key words** *Automatic control, Remote laboratory*

## Abstract:

*Education in the field of automatic control requires adequate practice on real systems for better and full understanding of the control theory. Experimenting on real models developed exclusively for the purpose of education and gaining necessary experience is the most adequate and traditionally it requires physical presence in laboratories where the equipment is installed. Remote access to laboratories for control systems is a necessary precondition and support for implementation of the e learning in the area of control engineering. The main feature of the developed system is support for the development, implementation and testing of user defined control algorithms on remote controller laboratory. User can define control algorithm in some conventional programming language and test it using this remote system.*

## 1 Introduction

Automatic control has become a major field in almost all engineering disciplines and automatic control courses are part of the respective engineering curricula. Control system education takes place in many different academic departments and disciplines, and control systems applications occur in wide variety of technologies. Control system science and engineering is concerned with automation. It involves a variety of tasks such as modeling, identification, simulation, decision making and optimization, coping with uncertainty through the feedback, and performance evaluation. In addition, successful application of control principles involves the integration of various tools from related disciplines, such as signal processing, electronics, communications, software, algorithms, real-time computing, sensors and actuators, as well as application of the specific knowledge. Application areas of control automation include transportation, manufacturing, communications, aerospace, process industries, and commercial products. The basic control systems principles influence and impact all of these application areas, as well as diverse fields of study such as biology, economics, and medicine.

Education in the field of automatic control requires adequate practice on real systems for better and full understanding of the control theory, as well as gaining some practical experience that is very important for the future work and development of control systems for real life processes in industry and other areas of commercial exploitation. Experimenting with such real systems that are used in real life is not always possible, it is available to small and limited group of people, it could be hazardous and might jeopardize normal operation of some important real life system. Experimenting on real models developed exclusively for the purpose of education and gaining necessary experience is the most adequate and traditionally it requires physical presence in laboratories where the equipment is installed. Extension of the traditional learning to e learning,

offers new possibilities for studying anywhere and anytime with respect to individual learning pace and capabilities. Remote access to laboratories for control systems is necessary precondition and support for implementation of the e learning in the area of control engineering. Functional requirements for the control engineering laboratory exercise consist of two parts. In the first part, real control model is setup in a way that is unknown to remote user. The remote user at first has a task to determine the characteristics of the real system model that is to be controlled. It is called identification of the system. User can perform the identification of the system by choosing the input signal to apply to the system. Input signal can be for instance step function, sinusoidal function or user defined function that consists of the array of consecutive values supplied by the user. User can setup parameters of input signals such as height of the step function, amplitude, offset, frequency and phase for sinusoidal function, array values for used defined function, and finally duration and sampling interval for the selected input signal. After that, user initiates the identification by applying the input signal to the system. The reaction of the system is measured and returned to the user in the form of numerical measured values that are also graphically presented in the form of the time function. User can repeat identification process as many times as necessary with different input signals and different parameters of input signals. After identification, the user is required to develop adequate control algorithm for the given system. In developing the control algorithm, user can choose from any of numerous existing controllers with no principal limitations. For easier selection and design of the control algorithm, a few standard controllers are available and implemented. To use any of these system supplied controllers, user has just to specify values for the controller parameters. Control algorithm developed by the user should be implemented in some programming language with given required interface. Strict respect of the controller interface ensures that the control algorithm can be included in remote system as replaceable software module. In that way, the user is isolated from the rest of the system and needs to concentrate only to controller design. Everything else is performed by the rest of the system that use control algorithm defined by the user. After implementing, the user uploads the file with control algorithm to the remote system, and starts the control process with his own controller implementation. In that way, the user is able to test his controller implementation on the real system model. That is the most important feature of this system, as the user can see in real, how effective is his implementation of the control algorithm. Depending on the output from the controlled system presented graphically and numerically, user can decide if the controller requires any change. Once the controller algorithm developed by the student obtains the final form, the program listing with controller implementation is uploaded to the system for review by the professor. This interactive process of development and implementation has some very important educational aspects. The user has to go through the whole process of the system identification, controller design, implementation, testing and eventual iterative process of repeating the system identification, controller redesign, implementation and testing. This remote system for controller development fits quite well into the existing general structure of the RemoteLab [1, 2] system, and is therefore implemented as additional hardware/software module of the existing RemoteLab system with no changes of the previous experiments. Remote Lab system was also upgraded, and now it became the main component of the Web Laboratory of University of Kragujevac (WLUK) [3]. (WLUK) was formed on the initiative of the group of teachers from the University of Kragujevac with the main aims of more efficient using of the equipment, development of laboratories [6,7] and strengthening of cooperation between departments, faculties and universities. Using of laboratory equipment with the web access is for authorized group of users with the purpose of academic education [4,5], research and permanent education. For the time being, WLUK consists of already existing laboratory equipment adapted for the web access and some specifically developed for web experiments [1, 2]. Some projects for the design of web based laboratory setups with appropriate accompanied e-learning contents financially support the WLUK, while some others are submitted and are in the process of evaluation. Web portal of WLUK is also combined and complemented with appropriate teaching materials and student services.

## 2 Structure of the system for experiments with automatic control

Experiment with automatic control further expands the RemoteLab system. It is an additional hardware / software module that is added to the system. Structure of the whole RemoteLab system is given on UML deployment diagram in figure 1.

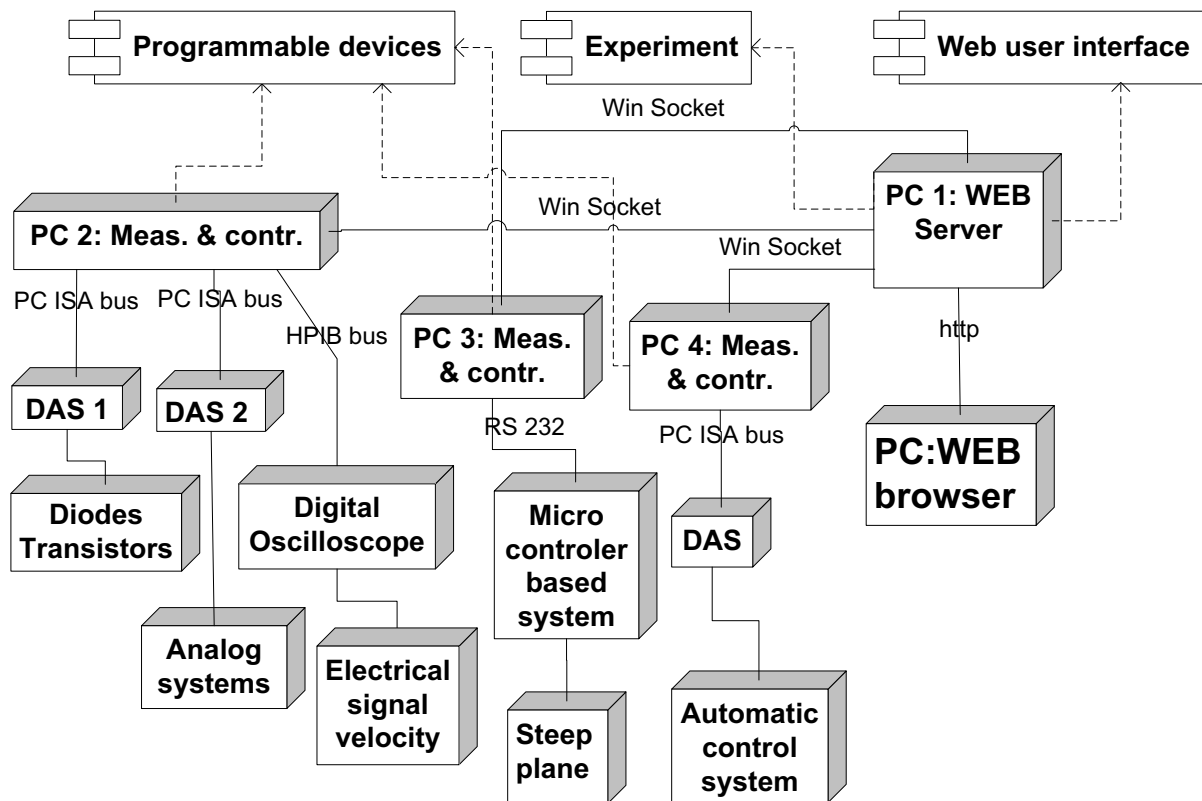


Figure 1 UML Deployment diagram for RemoteLab system

Automatic control system is attached to PC 4 using the Data Acquisition System (DAS). Although it is physically situated in different building from the rest of the system, network connection using Win Sockets with the web server makes it quite equivalent with other RemoteLab components. Software modules for support of this experiment are developed and implemented as classes in Web user interface component, Experiment and Programmable device drivers. This experiment fully fits into the existing RemoteLab architecture.

## 3 Experimental setup

Node named Automatic control system in figure 1 that is attached to PC 4 via DAS, contains hardware for experiments with automatic control. The main component is physical simulator, the electronic circuit that consists of three independent units. The first unit can represent or behave as the integrator, second unit represents linear system of the first order, and the third unit represents the linear system of the second order. Transfer functions of those linear systems are given as:

Transfer function of the integrator

$$W_b(s) = \frac{U_{iz}(s)}{U_{ul}(s)} = \frac{0.92}{s} \quad 1)$$

Transfer function of the linear system of the first order

$$W_c(s) = \frac{U_{iz}(s)}{U_{ul}(s)} = \frac{K}{Ts+1} \quad 2)$$

Transfer function of the linear system of the second order

$$W_a(s) = \frac{U_{iz}(s)}{U_{ul}(s)} = \frac{b_1s+1}{\frac{1}{a_2}s^2 + a_1s+1} \quad 3)$$

Figure 2a shows PC 4 with DAS and physical simulator, and figure 2b shows photograph of the physical simulator.

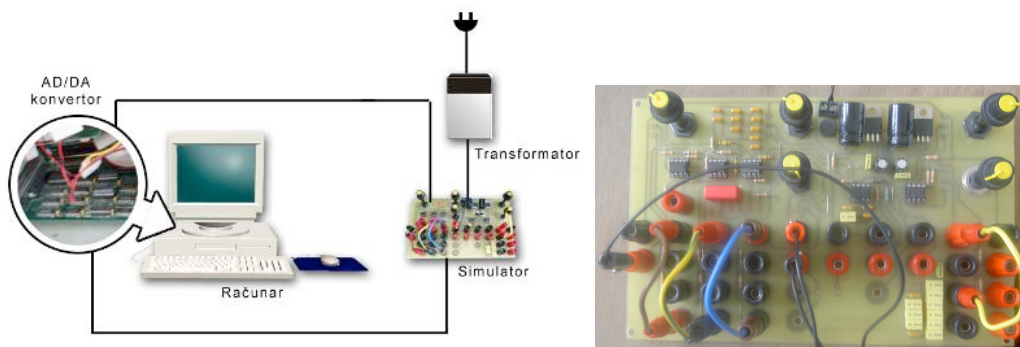


Figure 2a left, PC 4, DAS and physical simulator; 2b right, photograph of the physical simulator

Physical simulator can simulate typical industrial processes and can be setup using jumpers and potentiometers. Mentioned independent units of the physical simulator with the transfer functions 1, 2 and 3 can be connected in cascade, thus forming the system of the third order if desired. Zero of the transfer function can be adjusted to be in the left or right  $s$  plane, thus forming the system with minimal or non minimal phase characteristics. As the voltages of the physical simulator are limited to 10 V, and also the operating ranges of the AD and DA converters are limited to 9V, non linear processes of the type of saturations are possible in the system. Used DAS is 12 FPC-010 Flytech, 12 bit with 8/16 input channels AD and one channel output DA.

## 4 Experiments with control system

The experiments with automatic control systems are an important support for subjects Design of automatic control systems and Computer supported measurement and control on the Mechanical Engineering faculty. On the Faculty of Science, department of Physics, it is used in subjects Electronics and Data acquisition. It can also be used for similar subjects on other technical faculties in the fields of linear systems, automatic control, electronics, data acquisition. The experiment allows for the student to apply acquired theoretical knowledge from the fields of electronics, data acquisition, system identification, signal processing, system analysis and synthesis, control systems implementation and real time programming. Experimental setup consisting of PC with DAS and physical simulator can be presented with the block diagram in figure 3.

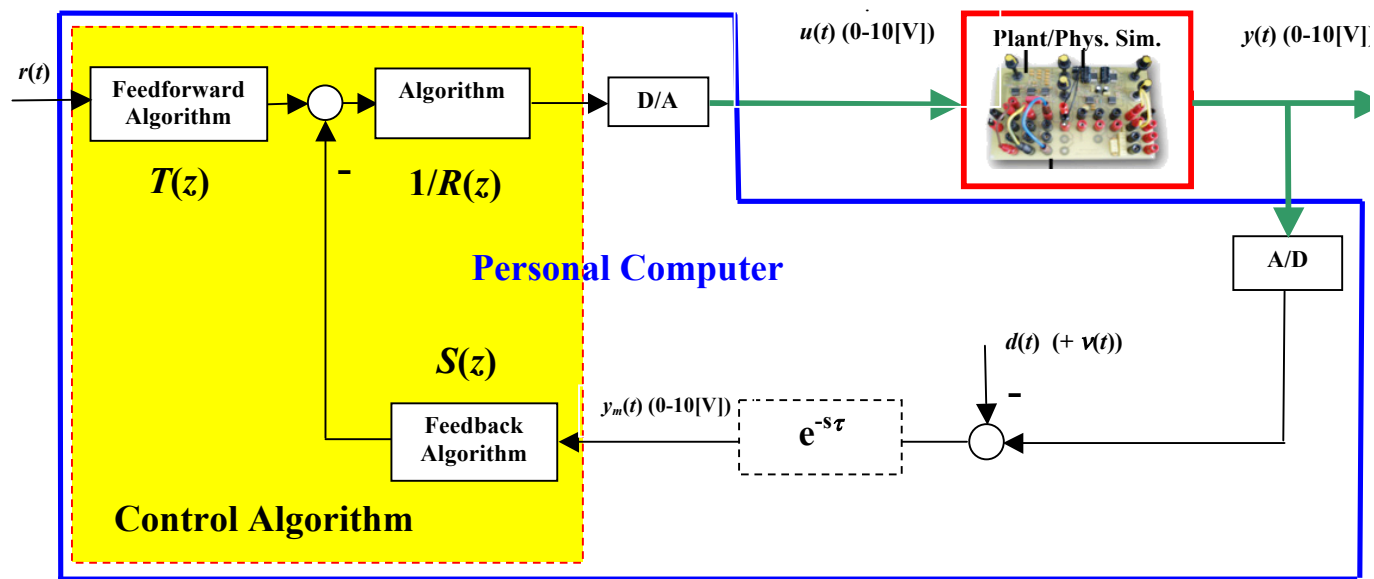


Figure 3 Block diagram of experimental setup

Blue line border in figure 3 encloses hardware and software modules in the PC. Red line border encloses physical simulator that is outside the PC. Hardware modules inside the PC are AD and DA, while software modules are enclosed with dashed lines. AD and DA are used for connection with physical simulator. Software modules inside the yellow surface belong to controller and are implemented, while the only software module outside the yellow surface with transfer function  $e^{-s\tau}$  represents the transport delay  $\tau$  and it is not yet implemented. Input signal  $r(t)$  on the left is the signal that the linear system – physical simulator has to follow in time as close as possible. In ideal case, the output signal from the physical simulator  $y(t)$  is identical to  $r(t)$ , which is never actually the case. The main task of the controller is to provide that the output signal  $y(t)$  is as close as possible to the input signal  $r(t)$ . Experiment can be divided in two main phases. The first phase is a system identification, and the second phase is controller design, implementation and testing.

#### 4.1 System identification

Parameters of the physical simulator – coefficients of the corresponding transfer function can be changed and set up with potentiometers. Also, the transfer function order of the physical simulator can be changed among the first, second and the third order. Student accessing the experimental system from the web has no knowledge of the selected order and parameters of the physical simulator. The first task is to properly identify the system – physical simulator, which means to determine the order of the physical simulator and parameter values – coefficients of the physical simulator transfer function. Identification can be performed by analysis of the responses of the system under consideration to selected input signals. Figure 4 shows the condensed web user interface for experiment with automatic control.

**Control system setup**

[Web Laboratory](#) > [Control System](#) > Control system setup

Input signal selection

☒ Step function      Step function height (V)   
 Amplitude (V)   
 Offset (V)   
☐ Sinusoidal function      Frequency (Hz)   
 Phase (deg)   
☐ User defined function      Definition file

☐ Controller

☒ P controller      P const.   
☐ User defined controller      Definition file

[Reset](#)

Measurement status:

☐ White on Black ☐ Horizontal axis ☐ Horizontal tics ☐ Vertical axis ☐ Vertical tics

Number of horizontal tics  Number of vertical tics

Image of the input function and system response      System response values in file

Figure 4 Web user interface for automatic control experiment

User can make the choice of the input signal among the step function, sinusoidal function and user defined function. Step function has three parameters that the user can control and these are “Step function height” (V), “Duration” (s) and “Sampling interval” (ms). Parameters for sinusoidal function are “Amplitude” (V), “Offset” (V), “Frequency” (Hz), “Phase” (deg), and also the “Duration” (s) and “Sampling interval”. In the case of user defined function, user uploads the text file with sequence of values that define the function. User selects the file which he prepares in advance by pressing on the “Browse” button, write below the “User defined function” option control. The only required parameter for user defined function is the “Sampling interval”. After selection of the input signal, identification is started with pressing the button “Start experiment”. Before the data from the web page are actually sent to the server, input values on the web form are checked with Java Script for eventual inconsistencies. It saves the time, as in the case of inconsistent input data, warning message is immediately shown, without sending request to the web server. If the input data are consistent, web browser sends selected data to the web server, which then sends the message using Win Sockets to PC for automatic control. Based on received data from the web server, PC for automatic control generates the required input function in real time and makes

measurements of the output signal from the physical simulator. In this part of the experiment, there is no feedback on physical simulator, which means that controller is not active as it is not defined yet. During the identification phase, the system can be presented with figure 5.

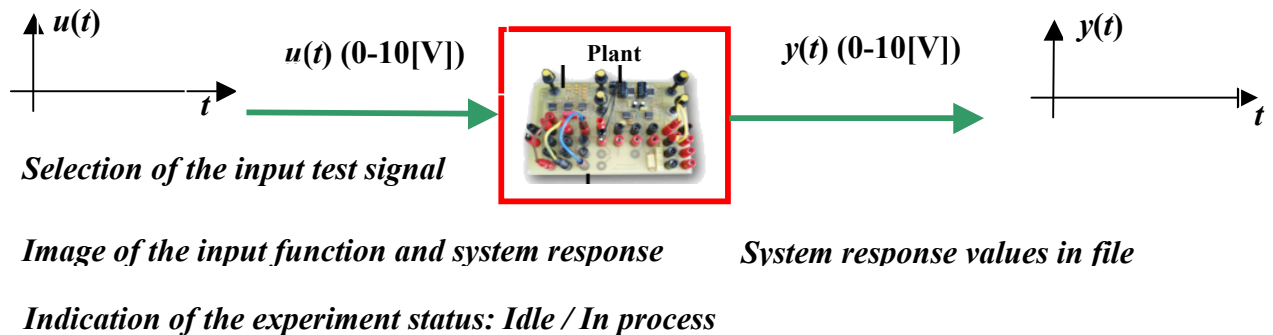


Figure 5 Simplified system function for identification phase

When the web server receives the measured output signal from the physical simulator, sent by the PC 4 for automatic control using win sockets, for the selected input signal, it creates the bmp file for graphical presentation of the input and output signal. It also creates text file with three columns for time, input signal and the measured output signal. That text file can be downloaded from the web site by the student. When the web server receives the data from the system, web user interface is changed, as the two new hyperlinks appear at the bottom. Plain text "Image of the input function and system response" and „System response values in file“, turn to blue color, as they become hyperlinks, as in figure 6.

**Control system setup**

[Web Laboratory](#) > [Who we are](#) > [Control system home](#) > Control system setup

Input signal selection

☒ Step function      Step function height (V)   
    Amplitude (V)   
☐ Sinusoidal function      Offset (V)   
    Frequency (Hz)     
    Phase (deg)   
☐ User defined function      Definition file

☐ Controller

☒ P controller      P const.   
☐ User defined controller      Definition file

[Reset](#)

Measurement status:

☐ White on Black ☐ Horizontal axis ☐ Horizontal tics ☐ Vertical axis ☐ Vertical tics

Number of horizontal tics  Number of vertical tics

[Image of the input function and system response](#)      [System response values in file](#)

Figure 6 New hyperlinks at the bottom of the web user interface, after measurement

Hyperlink “Image of the input function and system response” directs the web browser to the page displayed in Figure 7. Figure 7 shows the graphical presentation for the step function with height=5V, duration=5s and sampling time=100ms.



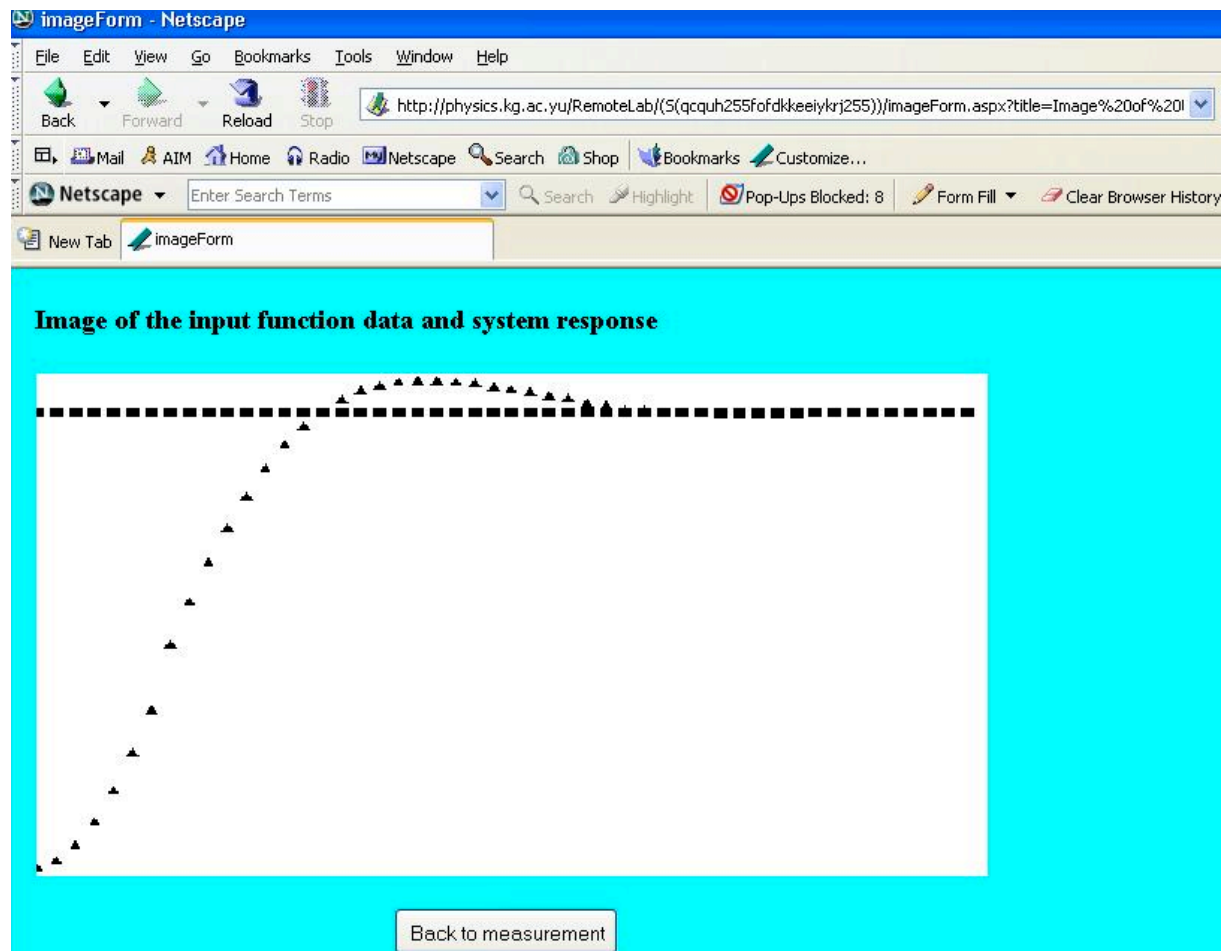


Figure 7 Graphical presentation for input selected on figure 4

Every measured value is represented by the small rectangular or triangle. Rectangulars correspond to the input signal which is the step function, and triangles correspond to the measured output signal. The other hyperlink, "System response values in file" directs the web browser to display text file with time values, values for the input function and measured values of the physical simulator output, arranged in three columns. Figure 8 shows the text file in the web browser.

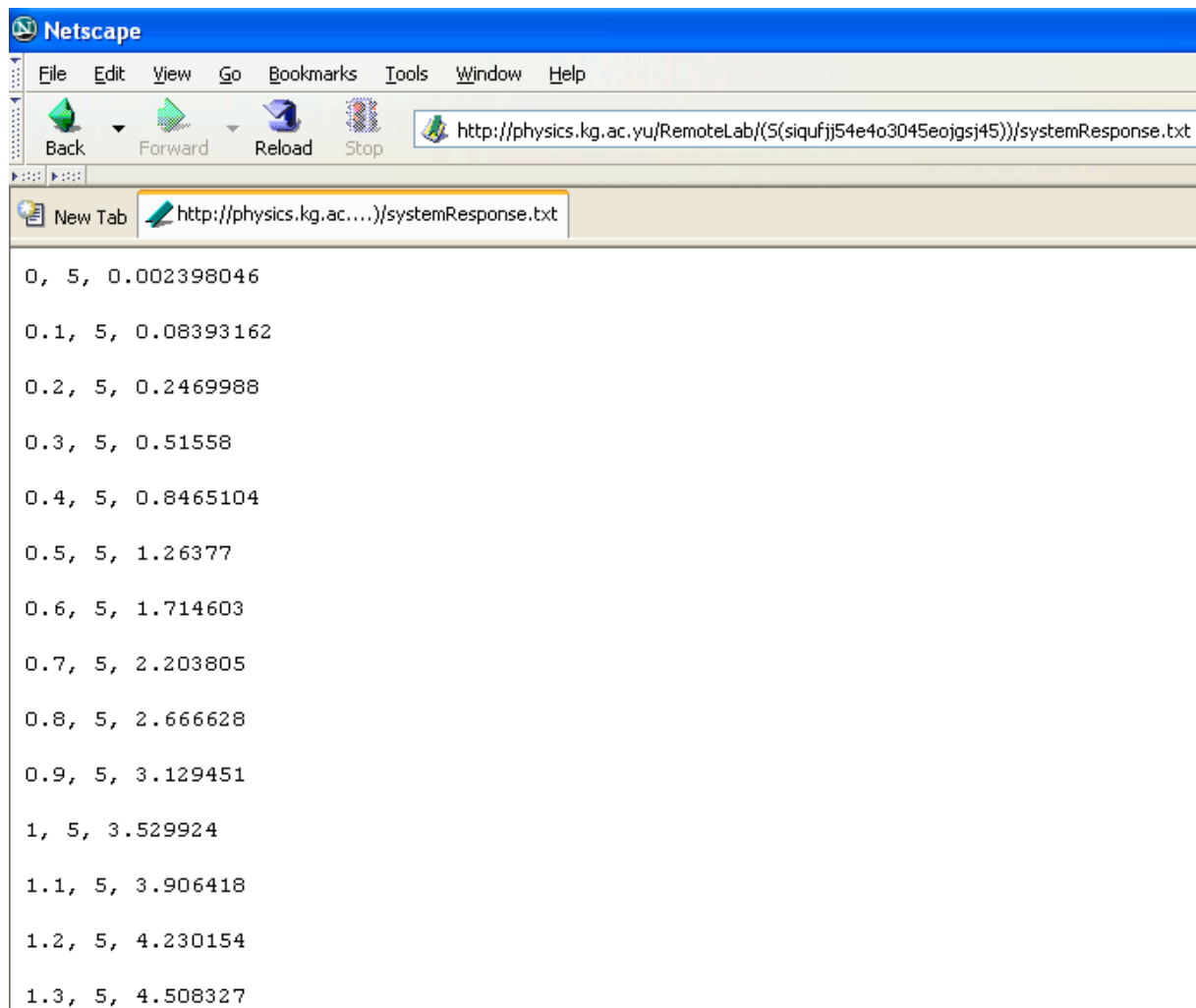


Figure 8 Text file with values for time (s), input signal (V) and measured output signal (V) organized in three columns

Check boxes at the bottom of the web user interface in figure 4 control the appearance of the image with graphical display of the measured data. User can select between black and white background, control the appearance of the vertical and horizontal axis, appearance of axis ticks and number of the ticks. Figure 9 shows the response of the physical simulator for sinusoidal input function, white on black background, horizontal and vertical axes with tic marks on the axis.

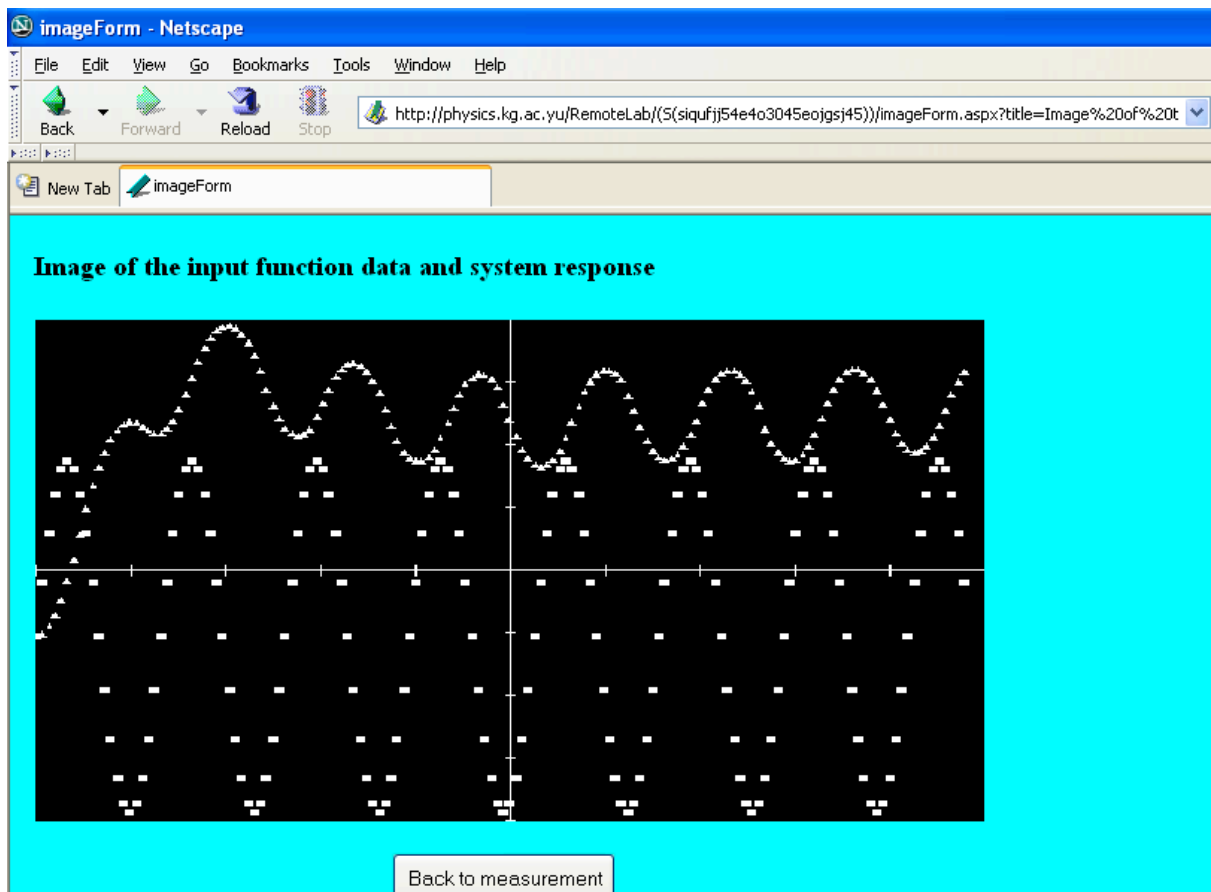


Figure 9 Graphical presentation for sinusoidal input function white on black with axis and ticks

Sinusoidal input function in figure 9 has amplitude=4V, offset=5V, frequency=0.5Hz, initial phase=0, duration=15s and sampling time=100ms. Rectangulars are input values and triangles are measured output values from the physical system. Example of the user defined function is given in figure 10.

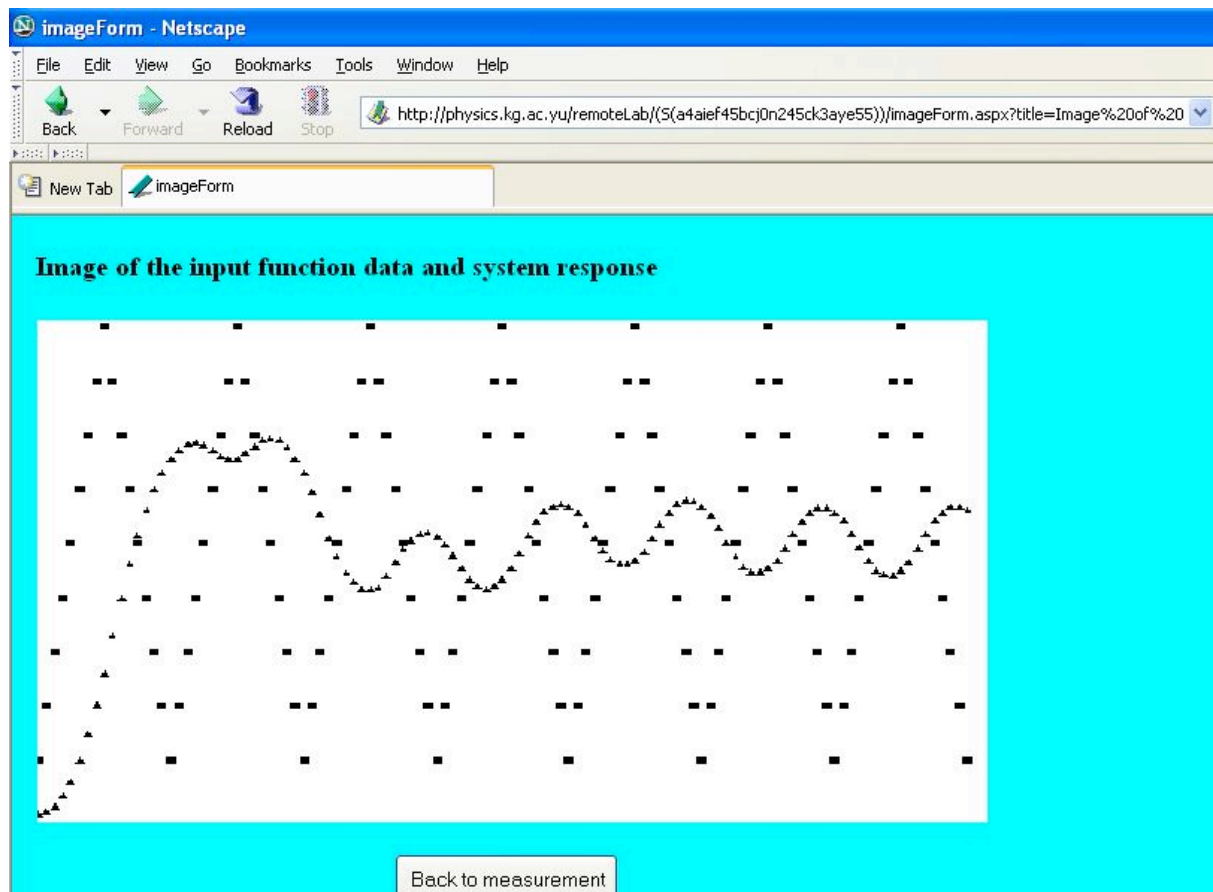


Figure 10 Graphical presentation for user defined function

User defined input function in figure 10 is of the form of the two linear functions which make triangles, with every value defined in the text file for the user defined input function. The only specified parameter for the user defined function is the time sampling-100ms for signals displayed in figure 10.

## 4.2 Controller design

After the phase with system identification is over, when the physical simulator order is known and the simulator parameters, the user can design the controller for the identified physical simulator. The controller is inside the yellow rectangular in the figure 3. It is defined in the form of algorithm, and implemented as the software module in some programming language. The structure of the control software on PC 4 that controls the DAS for automatic control experiments is modular, so that the module that defines the controller can be replaced with the different module that implements different type of controller. Necessary condition is that the interface of the module with the rest of the software is always the same regardless of the controller type. In order for the module that implements the controller to be fully replaceable at the run time, it is implemented as the dll file. As long as the user – author of the dll strictly sticks to the required interface, the dll can be used – called from the control software on the PC 4. The user defined dll file can be uploaded from the user interface web page that is shown in figure 4, by pressing the “Browse” button just below the “User defined controller” option control on the web form. The user can also select default controller that is P or proportional controller. That controller is already implemented in the system, and there is no need for the dll file upload. Only the proportional constant that determines the controller gain should be specified. Besides specifying the controller – default or user defined, it is also necessary to specify the input function, in the same way as for identification phase. By specifying the input

function and controller definition, the user can see how effective his controller is. Depending on the output signal obtained from the physical simulator and selected input function, the user can see if his implemented controller has the desired characteristics. Figure 11 shows the system response for the step input function and the P controller with  $\text{gain}=3$ .

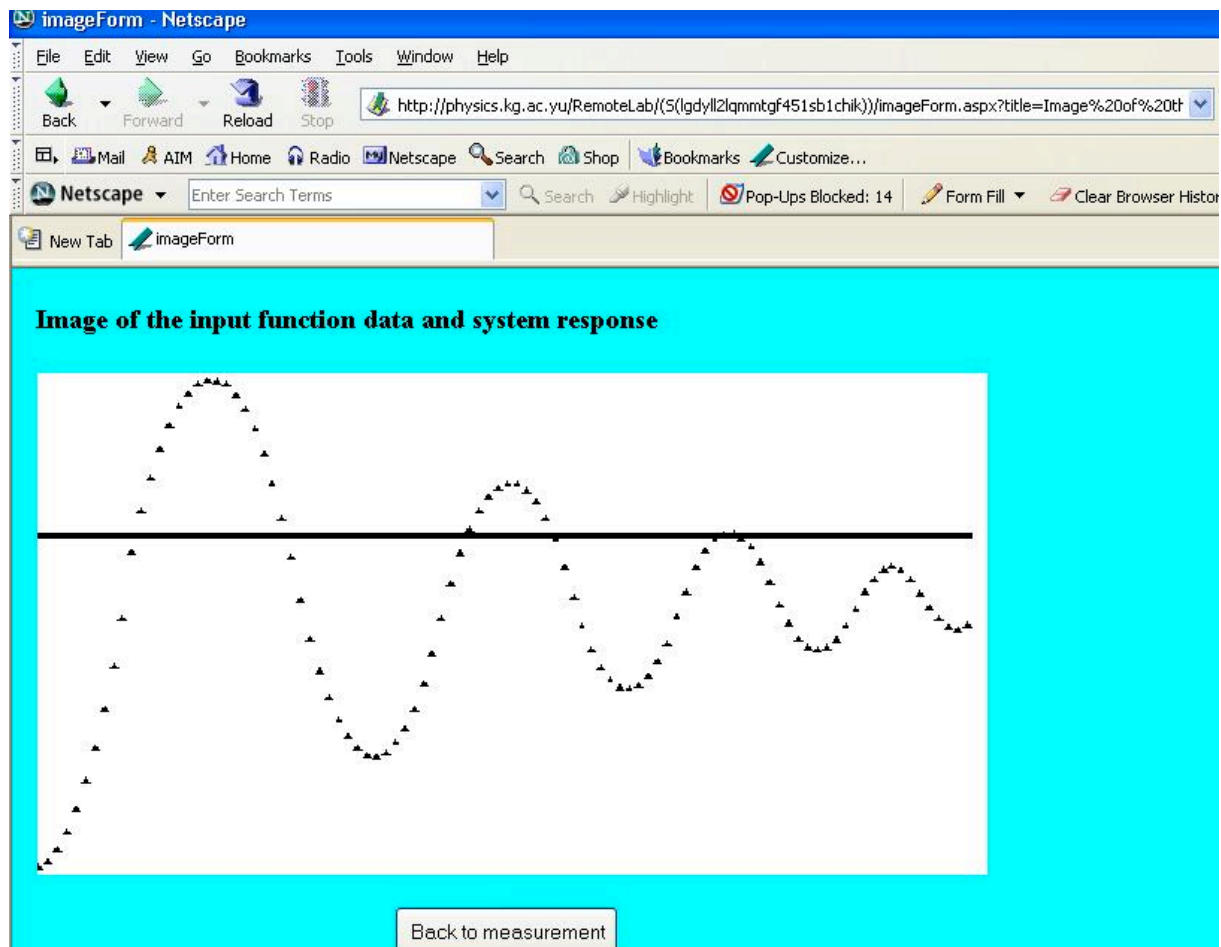


Figure 11 System response with P controller,  $\text{gain}=3$  with step input function

The same controller for different input signal, sinusoidal input function is given in figure 12.

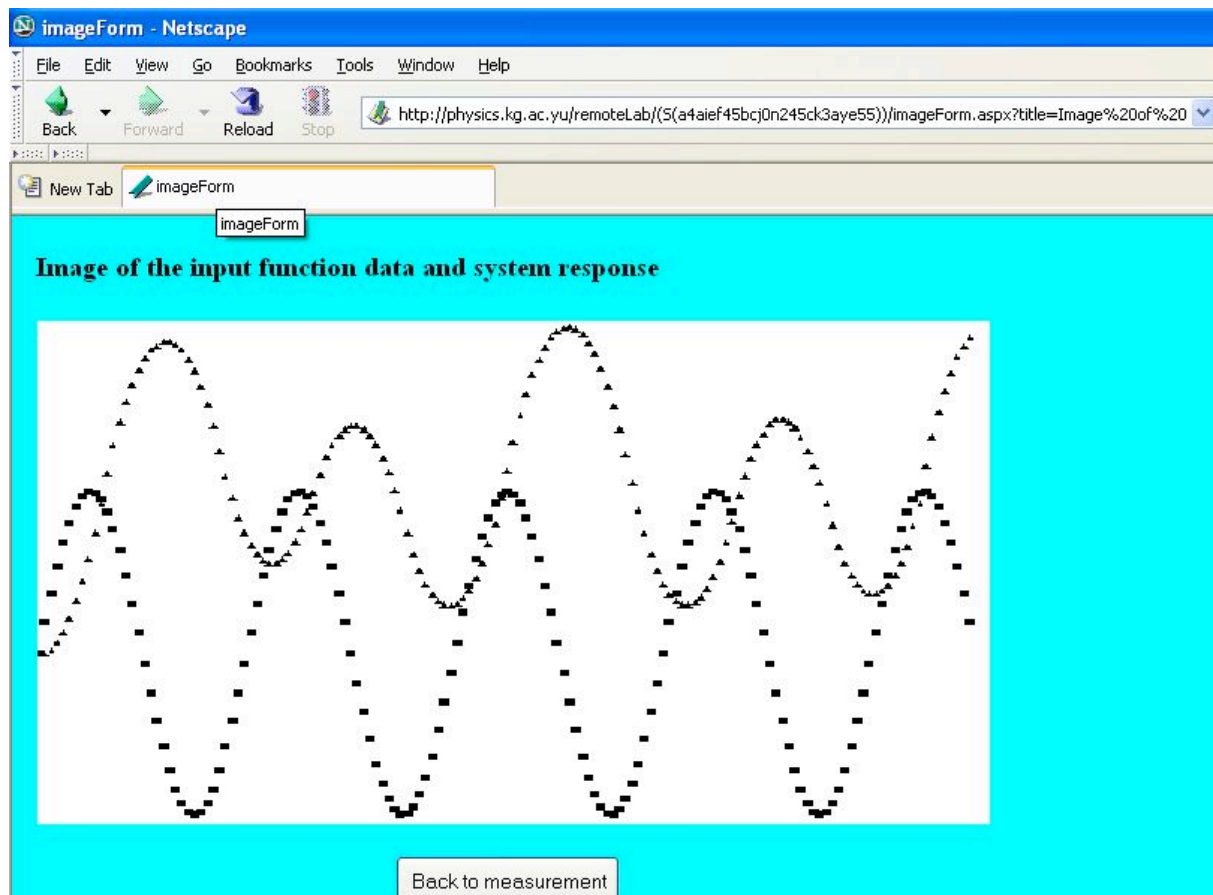


Figure 12 System response with P controller gain=3 and sinusoidal input function

## 5 Conclusion

Developed and implemented RemoteLab system became the main component of the WebLab of University of Kragujevac. Besides a number of experiments, Web Lab also has other teaching materials for e learning.

Experiments with automatic control are an important support for various subjects both on the Faculty of Mechanical Engineering and on Faculty of Science, as well as for other technical faculties. These experiments are performed on the module of the RemoteLab system for automatic control. Module of the RemoteLab system for automatic control consists both of hardware and software components. Hardware components are the physical simulator and PC with DAS. Software components are classes in three layers, for the web user interface, experiment and programmable device drivers. Physical simulator is a real electronic circuit that can behave as and imitate the typical industrial processes, allowing in that way the realistic and safe experimenting that resembles the conditions in real life systems that are frequently used in industry. Physical simulator has modular structure, with possibility of changing the parameters of the system. Experiments with automatic control require knowledge from various fields such as electronics, data acquisition, system identification, signal processing, system analysis and synthesis, control systems implementation and real time programming. The main aim of the developed system is to enable the student to practice tasks that he would perform on real systems in industry, and thus gain the necessary practical knowledge. Additional very important advantage of this system is that experimenting can be performed without physical presence in the laboratory, but instead, it can be performed only using the web browser, anytime, anyplace and according to individual capabilities. The system directly supports experiments that consist of two phases. In the first phase, user has to identify the real system represented with physical simulator. For identification user can select

among different input functions with variable parameters, and analyse the obtained response from the physical simulator. The other phase of the experiment is the design of the controller in the form of an algorithm, which is implemented in some programming language. System provides the user with the possibility to directly test the designed and implemented controller in the form of the dll which is uploaded as the replaceable software component. User can test the effectiveness of his controller for different input functions and in that way to test the controller in real different conditions. System provides the user with graphical and numerical presentation of the measured data for analysis of the controller behavior.

## References:

- [1] Yevgeniya S. Sulema, Vladimir M. Cvjetković, C.D, Remote Laboratory for Supporting e-Studies in Electronics, REV 2005, 30 June-1 July 2005 at "Transilvania" University of Brasov, Romania
- [2] V. Cvjetković, C D, RemoteLab system for real time experiments on WEB, YUINFO 2006, Kopaonik, 6 – 10 March 2006, Serbia
- [3] <http://weblab.kg.ac.yu>
- [4] M. Stefanović, M.S. Matijević, V.Cvjetković, "Web laboratories and engineering education", (invited paper in Serbian), XXXIII Quality Festival, Kragujevac, May, 2006, Serbia
- [5] D. Stevanović, A. Saranac and M. Matijević, "Client – server virtual laboratory model of coupled tank system", Second IFAC International Workshop IBCE'04, Grenoble, Sept. 2004, France
- [6] D. Gillet, "Web based experimentation: the will and the way", Second IFAC International Workshop IBCE'04, Grenoble, Sept. 2004, France
- [7] I.Benitez et al."Informatics technologies in automation laboratories", Second IFAC International Workshop IBCE'04, Grenoble, Sept. 2004, France