Remote Control Laboratory for Three-Tank Hydraulic System Using Matlab, Websockets and JavaScript [⋆]

Pavol Bisták [*]

[*] *Institute of Automotive Mechatronics*
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava, Slovakia
(e-mail: pavol.bistak@stuba.sk)

Abstract: This work aims to introduce a new architecture for building virtual and remote laboratories where the building blocks are represented by the Matlab/Simulink computing and simulation software, WebSocket communication technology and a front-end application created in JavaScript programming language. Matlab does not have direct support for WebSockets, but the implementation of the MatlabWebSocket library on the Matlab server has allowed connection through WebSockets that has been accepted with the client side realized in JavaScript. Additionally to the interactivity that is heavily supported by JavaScript, the remote laboratory has been visualized on the client side in 3D by implementation of the Three.js JavaScript library. From the control point of view the new remote laboratory enables to compare nonlinear feedback control with dynamical feedforward control respecting input saturation where in both cases a nonlinear disturbance observer can be used. WebSocket communication technology and the corresponding client interface in the form of a web application create possibilities for the presented remote laboratory to run from the Internet browser and no dedicated application is needed as it was in previous Matlab based laboratories what can be considered as a main contribution.

Keywords: hydraulic system, remote and virtual laboratory, Matlab/Simulink, WebSocket, 3D visualization, nonlinear control system

1. INTRODUCTION

The technological progress influences our lives in almost any area. The education is not an exception. At schools and universities we see many examples where digital technologies have a direct impact to the educational process. Engineering education has always been very progressive as concerning modern educational tools and has played a leading role in the development and implementation of new learning technologies. For the control engineering education it holds the same. The progress can be seen in extension of classical laboratory equipment (motors, electrical circuits, pendulums, magnetic levitation, temperature systems, etc.) to currently popular systems like drones, segways, RC cars and others.

In this contribution we concentrate on the hydraulic system that belongs to the classical systems but we present it in the modern form via virtual and remote laboratory. The hydraulic system with several tanks can be found in many places where control theory is taught. Its main advantage is the natural clarity of a process dynamics and a simple mathematical representation of behavior changes. Therefore a lot of universities exploit hydraulic systems very frequently to explain the basics of control theory (Hou and Zhu, 2013; Belikov and Petlenkov, 2014; Huang and Sira-Ramirez, 2015; Králik and Žáková, 2015). Our university was no exception when we developed a Matlab based software tool to control several hydraulic system configurations (Huba and Halás, 2011; Bisták et al., 2015; Bistak and Huba, 2016; Bisták and Huba, 2017). This paper describes the conversion of this tool to the modern interactive form given by web environment and from the theoretical point of view one can find its extension in the implementation of a previously designed nonlinear control law for the three-tank system respecting input signal constraints (Bisták, 2018). The presented nonlinear feedforward controller considers input saturation and is able to eliminate disturbances with the help of a nonlinear disturbance observer included in the control circuit. Moreover, properly designed filters can cause noise attenuation (Huba, 2015).

It is more than twenty years when we got first experiences with virtual and remote laboratories. In that time one could hardly imagine that today we would be browsing through virtual and remote laboratories using our mobile phones with high graphical resolution and fast operating processors that are powerful enough to support 3D graphics. Our developed online laboratory proves that this could be reality nowadays. Continuous development of virtual and remote laboratory technologies causes that they a becoming an integral part of our educational process and for "digital natives" they will be more and more popular.

[⋆] This work has been partially supported by the grants APVV SK-IL-RD-18-0008 and VEGA 1/0745/19.

Students like to prove new knowledge by experimentation and so they can perform the method ”learning by doing”. On the other side students feel free when experimenting via virtual and real laboratories because these should be built safely and student should be not afraid to damage the real laboratory equipment. Therefore, virtual and remote laboratories are today widespread and we can find many architectures and methodologies how to built and used them ((Heradio et al., 2016; Dormido et al., 2008; Duro et al., 2008; Sáenz et al., 2015; Chacón et al., 2015; Kalúz et al., 2015; Krbeček and Schauer, 2015; Magyar and Žáková, 2010)). From the technological point of view the novelty of the laboratory presented in this paper resides in the fact that the laboratory uses a new architecture based on the Matlab/Simulink software and WebSocket technology that allows to access the remote laboratory using just web browser. It has been firstly published in (Bisták, 2019) and this paper extends the possibilities of the WebSocket technology for more sophisticated control tasks. The contribution from the control theory point of view lies in the implementation of the dynamical feedforward controller to the nonlinear plant with input disturbances and noise.

The paper is organized as follows. Section 2 describes the model of the three-tank hydraulic system. Section 3 briefly outlines the controller design. Section 4 is focused on the 3D modeling of the three-tank system. Section 5 describes the new architecture of the designed remote laboratory. Section 6 deals with running an experiment. The paper results are then concluded by discussion in Conclusions.

2. THREE-TANK HYDRAULIC SYSTEM DESCRIPTION

In control engineering education the three-tank system (Fig. 1) is used frequently to demonstrate the principles of nonlinear systems. As the name says the main part of the system is created by the three tanks that in our case have a cylindrical shape. Another big tank called reservoir saves the liquid. Each tank has its own pump that serves for the transport of the liquid from the reservoir to the tank. Each tank has also its outlet with the valve that can be open or closed to control the output from the tank to the reservoir. There are two additional valves that control the flow between the first and second tank or the second and third one, respectively.

Denotations of variables and parameters can be seen from the schematic picture of the three-tank system (Fig. 2). In the first tank the variable h_1 denotes the liquid level height, the variable q_1 stands for the liquid inflow, the parameter A_1 represents the tank cross-section area and the parameter c_1 corresponds to the outflow valve coefficient. Similar sets of denotations h_2, q_2, A_2, c_2 and h_3, q_3, A_3, c_3 are valid for the second and third tank, respectively. The parameter c_{12} characterizes the valve between the first and second tank. Similarly, the parameter c_{23} corresponds to the valve between the second and third tank.

To express the dynamical behavior of the system the mass conservation law should be applied. It yields the system of three nonlinear differential equations

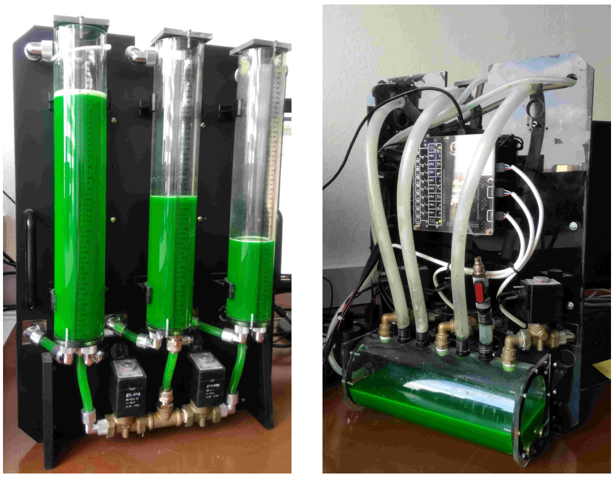


Fig. 1. Three-tank laboratory system - front and back view

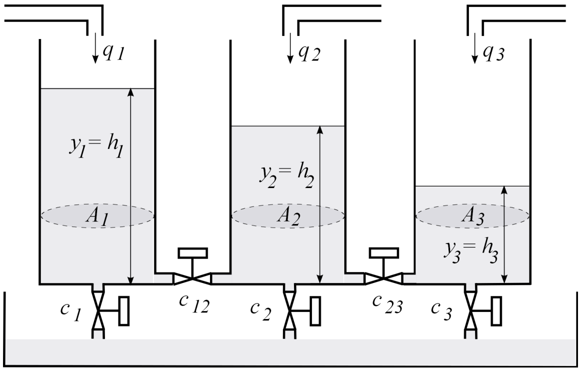


Fig. 2. Schematic drawing of the Three-tank laboratory system

$$\begin{aligned}
\dot{h}_1 &= \frac{1}{A_1} q_1 - c_1 \sqrt{h_1} - c_{12} \sqrt[\pm]{h_1 - h_2} \\
\dot{h}_2 &= \frac{1}{A_2} q_2 - c_2 \sqrt{h_2} + c_{12} \sqrt[\pm]{h_1 - h_2} - c_{23} \sqrt[\pm]{h_2 - h_3} \\
\dot{h}_3 &= \frac{1}{A_3} q_3 - c_3 \sqrt{h_3} + c_{23} \sqrt[\pm]{h_2 - h_3}
\end{aligned} \tag{1}$$

where the symbol $\sqrt[\pm]{\cdot}$ means

$$\sqrt[\pm]{z} = \begin{cases} \sqrt{z} & \text{if } z \geq 0 \\ -\sqrt{-z} & \text{if } z < 0 \end{cases} \tag{2}$$

Because of pump power limits the input values $q_i, i = 1, 2, 3$ are constrained $q_i = sat(\cdot)$, i.e.

$$Q_{min} \leq q_i \leq Q_{max} \tag{3}$$

Beside the power of each pump a computer can control each valve. By opening or closing the valves different configurations of the hydraulic system can be set up. To demonstrate the control algorithms designed in this paper the single-input single-output system of the third order is required. This means just the first pump will be powered (i.e. $q_2 = q_3 = 0$) and from the first tank the liquid will flow through the second and third one to the reservoir while the outlet valves of the first and second tank will be closed (i.e. $c_1 = c_2 = 0$). From the measurements on the real hydraulic system the following values have been identified: $A_1 = 1 \cdot 10^{-3} m^2$, $c_{12} = 1.48 \cdot 10^{-2} m^{\frac{1}{2}} s^{-1}$, $c_{23} = 1.52 \cdot 10^{-2} m^{\frac{1}{2}} s^{-1}$ and $c_3 = 6 \cdot 10^{-3} m^{\frac{1}{2}} s^{-1}$. According to the maximal value of the first pump the constraints of the control value q_1 have been determined as $Q_{min} = 0\, m^3 s^{-1}$ and $Q_{max} = 1 \cdot 10^{-5} m^3 s^{-1}$. The controlled value will be the height of the level in the third tank (i.e. output $y = y_3 = h_3$).

3. DESIGN OF DYNAMICAL FEEDFORWARD CONTROL WITH INPUT SATURATION AND DISTURBANCE REJECTION

The above described nonlinear system is going to be controlled by a nonlinear feedforward controller respecting input saturation. The task of the nonlinear controller will be to linearize the nonlinear system first. For this purpose the exact linearization method will be used that is able to fully linearize the system. After getting the linearized system another nonlinear controller will be applied - the controller that respects control signal constraints (3). After that as the resulting controller has the form of a PD-controller an integral part of the control must be added to cope with disturbances. Therefore a nonlinear input disturbance observer will be designed. The last part of the control design consists in dynamical feedforward control that exploits all the previously designed controller parts to construct the control system that can get over time delays. For more details, a reader is referred to Bisták (2018).

4. MODELING OF THREE-TANK SYSTEM IN 3D

An interaction with the system described in the previous sections will be based on communication via the Internet. In order to provide a user with the most authentic output of a real-time experiment running in a remote laboratory, the appropriate representation of received data by the client side is desired. Since human cognition is highly dependent on visual input, a 3D visualisation has been considered the proper solution. The 3D model of the three-tank system not only meets the stated requirements, but also contributes to a deeper user understanding of problems which may arise while performing an experiment. The remote user can inspect the construction of the system, its functionality, and dynamical behavior.

There are a few quality 3D modelling software products on the market at the moment, some of which, such as Blender, are open source. Despite having these options, the commercial 3D Studio MAX still remains a broadly used tool amongst professionals, for it enables them to create realistic models and fluid animations. Therefore, the academic version of the 3D Studio MAX has been exploited to create a 3D virtual representation of the real-world three-tank system. The process has begun by arranging simple geometrical objects resembling only the most important parts. However, the final design comprises all essential components precisely constructed to match the measures of the original system (Bisták, 2019).

This 3D model has been subsequently incorporated into the method of displaying data on the client side, where it is being rendered thanks to the Three.js JavaScript library. The library uses the standardized WebGL technology for manipulating and creating 3D objects while harnessing the computational power of graphic adapters. Its popularity amongst web developers stems from a decent performance and the fact that it offers a user-friendly API to utilize the already mentioned HTML5 technology, WebGL, without requiring the knowledge of this language and complex mathematical formulas. The final version of the three-tank system model captured directly from the Internet browser can be observed in Figure 3 (front view) and Figure 4 (back view).

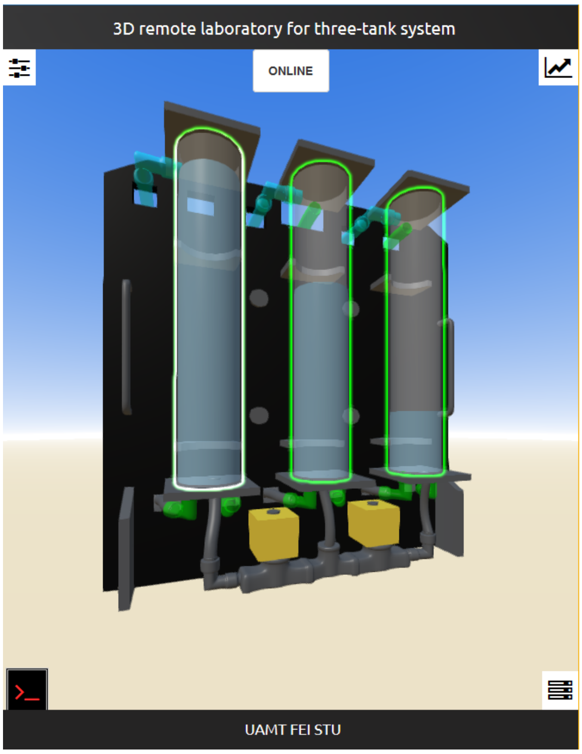


Fig. 3. 3D model of hydraulic system - front view

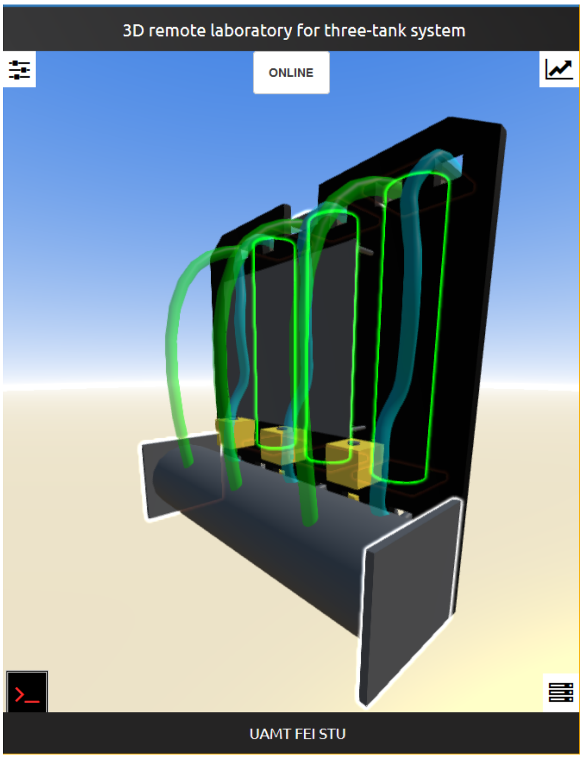


Fig. 4. 3D model of hydraulic system - back view

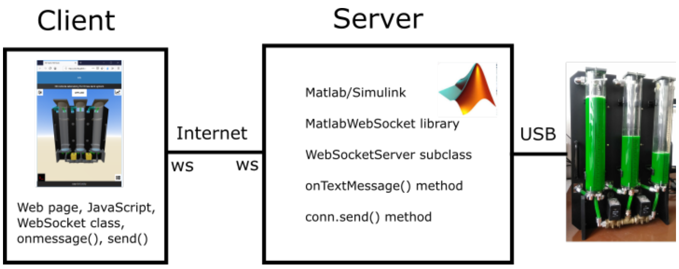


Fig. 5. Client-server architecture of remote laboratory

5. REMOTE LABORATORY BASED ON MATLAB AND WEBSOCKETS

This part can be seen as a continuation of the work started in Bisták (2019) and extends the possibilities of the WebSocket technology for more sophisticated control tasks.

The client-server architecture is perceived of as being a good practice in the matters of communication with a remote laboratory. Thus, it has been chosen for our purposes too, as shown in the Figure 5. After establishing a connection between a server and a client, the initial message containing parameters of an experiment is sent to the server over the Internet. Consequently, the server starts the experiment with the received parameters and begins to simultaneously stream the acquired data back to the client. The user can then examine the results of the experiment in real time both in the form of graphs and an animation of the previously described 3D model.

In order the server, based on the Matlab software, could control the physical three-tank system, the USB interface has been used to connect these two subjects of the remote laboratory. The data obtained throughout an experiment are being simultaneously transmitted to a client via WebSockets. In comparision with Web services (e.g., REST) this technology retains a permanent connection between the server and the client and facilitates more effective communication by lowering the necessity of requesting for each piece of data individually. The server sends a message as soon as it gathers new information, which eliminates any redundant network traffic. Nonetheless, a rising number of existing connections may become a cause for concerns.

While JavaScript provides native support for WebSocket communication technology, Matlab offers no such solution on its own, which has implied embodying a third-party library, MatlabWebSocket (Jebej, 2016), in our implementation. This library consists of two parts: a WebSocket server and a WebSocket client for Matlab. The WebSocket server is based on an implementation of the WebSocket protocol in Java. To implement the WebSocket server, it is necessary to define its subclass. This subclass should implement the following methods: *onOpen()*, *onTextMessage()*, *onBinaryMessage()*, *onError()* and *onClose()*. The arguments of these methods are, in addition to the object instance itself, *message* - a message received by the server and *conn* - an object representing the client WebSocket connection. For our purposes, the most interesting method will be the *onTextMessage()* method and its *message* argument that allows the server to receive parameters data coming from the client. Also the *conn* argument is important as it will enable to identify the client and send to it back e.g. measured data through the object *conn* using its *send()* method. Of course, it is also necessary to implement WebSocket communication on the remote client side, but this is not a problem, because there is an API that establishes WebSocket as a permanent connection between the client and the server. Thus the client can initialize a connection from a web page by sending a message through WebSockets and this event can be recognized by the server's *onTextMessage()* method that can extract required parameters from the *message* argument and identify the client's connection according to the *conn* argument.

After the server has received the experiment parameters, the corresponding block diagram is started in Simulink (Fig. 6). In this diagram, the 3-tank system block represents a real hydraulic system that is connected via a USB cable to a server PC running Matlab with an instance of the WebSocketServer subclass. Alternatively, it can represent the Simulink model of the hydraulic system in the case of virtual laboratory. Also the type of controller can be switched between feedforward and feedback. There is one block in the block diagram (the Interpreted Matlab Fcn block highlighted by the orange color) that calls the function responsible for sending the measured data from the experiment back to the client via the same WebSocket connection. This is done by the *conn.send()* method, which, thanks to the *conn* object, can respond to the client from which the experiment parameters were received. As a result, data between Matlab and the JavaScript client can be transmitted in both directions, supporting dynamic visualization of the three-tank hydraulic system in 3D on the web page.

Fig. 6 represents the Matlab/Simulink realization of the control circuit with input saturation and disturbance observer described in the chapter 3. Beside the 3-tank system block (in blue) corresponding to the controlled system, the Feedforward/Feedback Controller block (in magenta) is displayed that includes the exact linearization and the PD-type controller respecting constraints. The Disturbance Observer & Filters block (in green) provides an integral part of control and ensures the disturbance rejection. In the middle of the above mentioned blocks one can notice the Saturation block (in cyan). The input signal blocks (Reference signal and Input Disturbance) are highlighted by the yellow color. Other non-colored blocks serve for data display and storage.

6. RUNNING THE EXPERIMENT

Students can access the Web page of the remote laboratory through the Moodle e-learning system. The access is limited because of hardware resources (at this time we possess just one hydraulic system therefore Moodle Booking or Reservation plugins could be advantageous) and also because of Matlab licensing. The initial Web page of the remote laboratory shows the 3D model of the hydraulic system. After activating the Connection menu button placed in the bottom right position a student can connect to the remote laboratory and its status changes from *Offline* to *Online* (Fig. 7). In the Connection menu the *Server url* field shows the address and the port of the

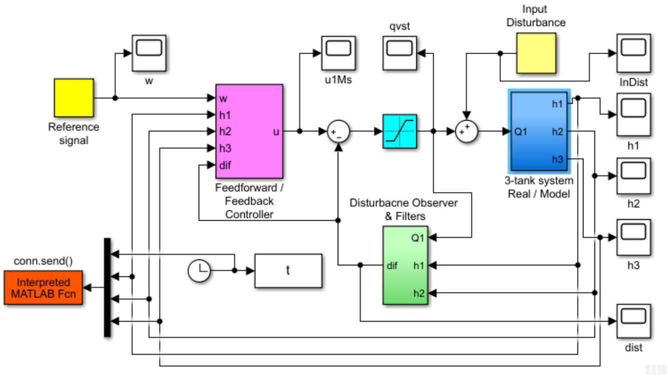
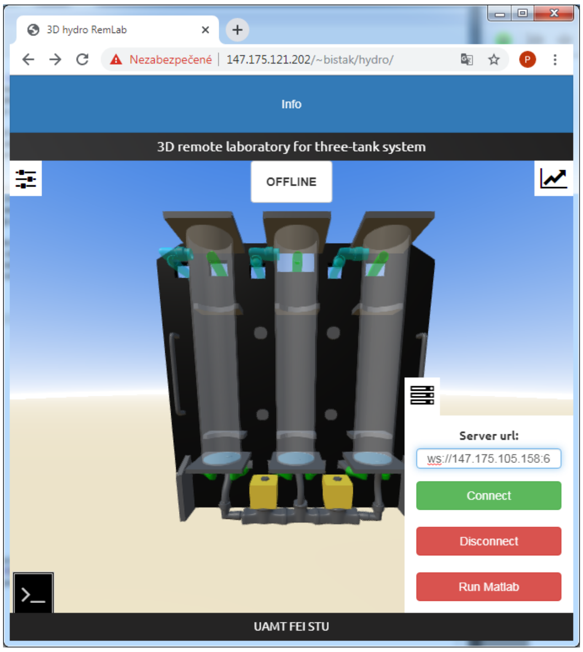
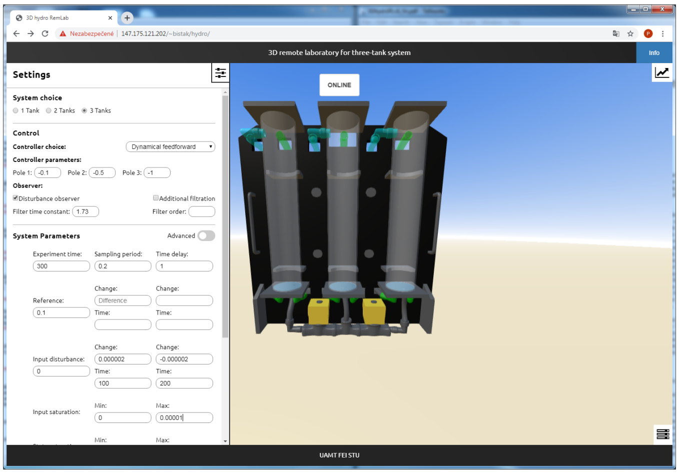


Fig. 6. Simulink block diagram of control circuit running at the server side of remote laboratory



Server url: ws://147.175.105.158:6

Fig. 7. Initial Web page of the remote laboratory with activated Connection menu


3D hydro RemLab
Nezabezpečené
147.175.121.202/~bistak/hydro/
3D remote laboratory for three-tank system
Info
ONLINE
UAMT FEI STU

Settings

System choice

○ 1 Tank ○ 2 Tanks ● 3 Tanks

Control

Controller choice: Dynamical feedforward ▾

Controller parameters:

Pole 1: -0.1 Pole 2: -0.5 Pole 3: -1

Observer:

☑ Disturbance observer ☐ Additional filtration

Filter time constant: 1.73 Filter order:

System Parameters Advanced

Experiment time: 300 Sampling period: 0.2 Time delay: 1

Reference: 0.1 Change: Difference Time: Change: Time:

Input disturbance: 0 Change: 0.000002 Time: 100 Change: -0.000002 Time: 200

Input saturation: Min: 0 Max: 0.00001

Min: Max:

Fig. 8. Setting the paremeters of the experiment

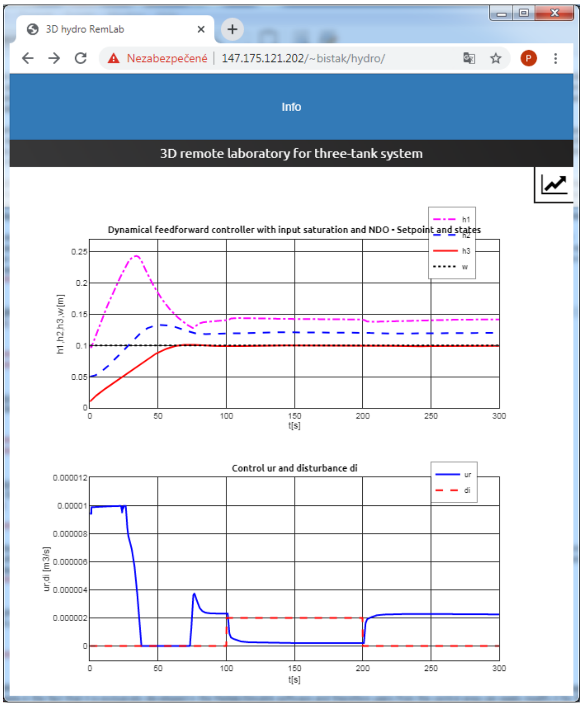


Fig. 9. Results of experiment in the form of graphs. Output and control time responses of the control circuit with dynamical feedforward controller.

Matlab server. After being connected the user can check the console of the remote laboratory placed in the bottom left corner for possible error messages. But usually the user continues with activation of the Settings button placed in the upper left corner (Fig. 8).

In the Settings menu the student can choose the number of tanks that will be used in the experiment. Then Control (where controller type can be switched between Dynamical Feedforward and Feedback) and Observer parameters could be set. The last section in the Settings menu is reserved for System parameters (experiment time, sampling period, time delay, reference, disturbance, input and state saturation). Using the *Advanced* switch the user can set the additional set of parameters (initial values, coefficient of valves, cross-section areas). After setting all parameters the experiment can start by pressing the *Run experiment* button placed in the bottom part of the Settings section.

According to settings from the Fig. 8 the client starts receiving the experimental data gradually. The changes are visible through the animation of the 3D model or animated graphs that are accessible through the Graph button placed in the upper right position. After finishing the experiment the receipt of data is stopped that causes the 3D model stops animation and also the graph plotting will be stopped (Fig. 9). The user can change the parameters and repeat the experiment or disconnect from the remote laboratory.

The results displayed in the Fig. 9 come for the selected dynamical feedforward controller from the experiment in the virtual laboratory mode. We do not present results of the real system because of some hardware imperfections (interconnection valves failure) we were limited to the operation of only the first tank that is not very interesting from the control point of view.

7. CONCLUSIONS

This paper presents a new architecture suitable for virtual and remote laboratories. The architecture is based on the WebSocket communication technology when a persistent connection between a client and a server is established. This type of communication enables on the client side to use a web user interface and design a web application based on JavaScript that brings a lot of interactivity and animation effects for the end user. The possibility to realize even the 3D visualization with the support of the Three.js library has been presented. To implement the WebSocket technology on the server side we have had to cope with the fact that Matlab has no built-in support for it. The third party library called MatlabWebSocket had to be exploited in order Matlab could communicate through WebSockets. The web user interface on the client side belongs to the main advantage of the proposed solution in comparison with the previous ones that were exclusively based on Matlab/Simulink, so this or other dedicated applications were also required on the client side.

As an example for a remote laboratory architecture verification the three-tank hydraulic system has been chosen because of its natural clarity and ease of understanding. Beside the technique of its 3D visualization the nonlinear controller has been applied. The built remote laboratory enables to compare two nonlinear controllers - feedback or feedforward ones up to the third degree of the hydraulic system. The developed laboratory will be used in the Nonlinear Control Systems course of the master studies. To compare the results with other conventional control methods the laboratory should be supplemented by the quality criteria evaluation feature. In the future, the remote laboratory can be enhanced by video transmission with elements of augmented reality.

REFERENCES

J. Belikov and E. Petlenkov. Model based control of a water tank system. In *The 19th IFAC World Congress*, pages 10838–10843, Cape Town, South Africa, August 2014.

P. Bisták. Dynamical feedforward control of three-tank system. In Roberto Moreno-Díaz, Franz Pichler, and Alexis Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2017*, pages 134–141. Springer International Publishing, 2018. ISBN 978-3-319-74718-7.

P. Bistak and M. Huba. Three-Tank Virtual Laboratory for Input Saturation Control Based on Matlab. *IFAC-PapersOnLine*, 49(6):207 – 212, 2016. ISSN 2405-8963. doi: http://dx.doi.org/10.1016/j.ifacol.2016.07.178. 11th IFAC Symposium on Advances in Control Education ACE 2016.

P. Bisták, M. Huba, and I. Bélai. Comparing Constrained Model Based Controllers for Nonlinear Hydraulic Plant. In *IFAC Workshop Internet Based Control Education (IBCE)*, Brescia, Italy, 2015.

P. Bisták. 3D three-tank remote laboratory based on Matlab and Websockets. In *2019 5th Experiment International Conference (exp.at'19)*, pages 85–89, June 2019.

P. Bisták and M. Huba. Three-tank virtual laboratory for dynamical feedforward control based on Matlab. In *2017 19th International Conference on Electrical Drives and Power Electronics (EDPE)*, pages 318–323, Oct 2017. doi: 10.1109/EDPE.2017.8123223.

J. Chacón, H. Vargas, G. Farias, J. Sánchez, and S. Dormido. EJS, JIL Server, and LabVIEW: An Architecture for Rapid Development of Remote Labs. *IEEE Transactions on Learning Technologies*, 8(4):393–401, 2015. doi: 10.1109/TLT.2015.2389245.

R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, and S. Dormido. Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System. *IEEE Transactions on Education*, 51(1):35–44, 2008.

N. Duro, R. Dormido, H. Vargas, S. Dormido-Canto, J. Sánchez, G. Farias, F. Esquembre, and S. Dormido. An Integrated Virtual and Remote Control Lab: The Three-Tank System as a Case Study. *Computing in Science Engineering*, 10(4):50–59, 2008.

R. Heradio, L. de la Torre, and S. Dormido. Virtual and remote labs in control education: A survey. *Annual Reviews in Control*, 42:1 – 10, 2016. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2016.08.001.

Z. Hou and Y. Zhu. Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems. *IEEE Transactions on Industrial Informatics*, 9(4):2301–2309, Nov 2013. ISSN 1551-3203. doi: 10.1109/TII.2013.2257806.

Congzhi Huang and H. Sira-Ramirez. A flatness based active disturbance rejection controller for the four tank benchmark problem. In *American Control Conference (ACC), 2015*, pages 4628–4633, July 2015. doi: 10.1109/ACC.2015.7172058.

M. Huba. Filter choice for an effective measurement noise attenuation in PI and PID controllers. In *2015 IEEE International Conference on Mechatronics (ICM)*, pages 46–51, 2015.

M. Huba and M. Halás. Hydraulic plants for face-to-face training and remote experiments. In *2011 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 79–81, Oct 2011. doi: 10.1109/ICETA.2011.6112590.

Jebej. MatlabWebSocket. `https://github.com/jebej/MatlabWebSocket`, 2016. Accessed: 2019-02-11.

M. Kalúz, J. García-Zubía, M. Fikar, and L. Čirka. A flexible and configurable architecture for automatic control remote laboratories. *IEEE Transactions on Learning Technologies*, 8(3):299–310, 2015. doi: 10.1109/TLT.2015.2389251.

M. Králik and K. Žáková. Interactive WebGL Model of Hydraulic Plant. *IFAC-PapersOnLine*, 48(29):146–151, 2015. doi: 10.1016/j.ifacol.2015.11.228.

M. Krbeček and F. Schauer. Communication and diagnostic interfaces in remote laboratory management systems. *International Journal of Online Engineering*, 11(5):43–49, 2015. doi: 10.3991/ijoe.v11i5.4926.

Z. Magyar and K. Žáková. Using SciLab for building of virtual lab. In *2010 9th International Conference on Information Technology Based Higher Education and Training, ITHET 2010*, pages 280–283, 2010.

J. Sáenz, J. Chacón, L. De La Torre, A. Visioli, and S. Dormido. Open and Low-Cost Virtual and Remote Labs on Control Engineering. *IEEE Access*, 3:805–814, 2015. doi: 10.1109/ACCESS.2015.2442613.