Arduino based online laboratory platform for digital control systems analysis and design

Vladimir Mitić, Vladimir Sibinović, Snežana Đorđević, Boban Veselić

Abstract— **This paper presents a laboratory platform with remote access for exercising digital control systems analysis and design. The laboratory environment utilizes three types of standard control systems: positional servosystem, velocity servosystem and temperature regulation system. It also has online access through a standard Web browser that enables remote execution, visualization and data acquisition. Students can perform different variety of experiments, ranging from bachelor to master level studies. They can work independently or collaborate within a group with multiuser access to the platform. The laboratory platform is designed a way to be applicable to any regulation system with implemented communication protocols.**

Index Terms— **Laboratory platform, remote access, digital control theory, Arduino.**

I. Introduction

The previous two years, because of the pandemic outbreak, introduced remote learning to all levels of education. While this type of learning is not new, it was never before used at such an extent and for a large number of topics.

Remote laboratories were always popular in engineering education, but their popularity has grown rapidly in recent years. With the development of different technologies, e.g., Internet-of-things, more and more researchers turn to developing remote laboratories for different topics.

The results presented in this paper stemmed from the efforts, started a long time ago, trying to modernize old laboratory teaching platforms that were used to explain control system fundamentals to students. The possibility of remote access to the laboratory setup was a priority from a very beginning. Remote experiments offer more flexibility both for teachers and students, and some experiments were performed at the Faculty of Electronic Engineering in Niš more than 10 years ago [1].

Gaining experience in that field over the years, the authors started developing a new laboratory platform for teaching control system fundamentals. With flexibility and simplicity, as the main guiding principles, a new platform has been designed for teaching digital control systems analysis and design. Remote access to the platform requires from a student to only have a device with internet access. On the other hand, the platform can be built using off the shelf components that are inexpensive.

The paper is organized as follows. Section II presents the hardware system design, while Section III describes the software architecture of the system Section IV shows the educational possibilities and potentials of the realized laboratory platform and demonstrates its ease of use through some examples of laboratory exercises.

II. Hardware Architecture

Arduino Due development board has been chosen to be the key component of the laboratory platform. Arduino is an open-source development environment that has become ubiquitous in education, from the primary schools all the way to the universities. Arduino platform provides different inexpensive development boards, that can be used for teaching different topics like programming [2], robotics [3], physics [4], etc. Arduino Due has an Atmel SAM3X8E Arm Cortex-M3 microcontroller, which uses a 32-bit RISC processor operating at the maximum frequency of 84 MHz. This gives the platform adequate processing power and enables further development of the system. It also has plenty of general-purpose input/outputs (GPIOs) for connecting other parts of the system. The main disadvantage is that this system is 3.3V transistor-transistor logic (TTL) and requires logic level converters to work with components having 5V TTL.

In order to be able to create a set of different laboratory exercises, the three standard control systems have been implemented: positional and velocity servosystems using DC motors, and temperature regulation system using heating elements.

A. *Positional and velocity servosystem*

The positional and velocity servosystems were realized using Crouzet 82830010 DC motor that has the nominal power output of 33W. A gearbox with the ratio of 12.25 is coupled with the motor for the positional servosystem, for more accurate angular position control. An absolute magnetic encoder with 4096 increments per revolution is chosen for position measuring. These measurements are also used for the angular speed estimation of the motor shaft [5]. This was chosen as the most inexpensive solution with the highest resolution. Since the encoder is absolute, a lot of attention has been dedicated to detecting the zero-crossing condition, especially when operating on high speeds. The motor speed is estimated using simple Euler derivative approximation, which means that for small sampling periods the speed signal can be noisy. According to the predicted motor current requirements, Pololu VNH5019 motor driver has been chosen. A custom printed circuit board (PCB) was designed

Vladimir Mitić is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: vladimir.mitic@elfak.ni.ac.rs)
Vladimir Sibinović is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: vladimir.sibinovic@elfak.ni.ac.rs)
Snežana Đorđević is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: snezanadjordjevic@elfak.ni.ac.rs)
Boban Veselić is with the Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia (e-mail: boban.veselic@elfak.ni.ac.rs)

 ISBN 978-86-7466-930-3

and manufactured for easy component interconnection and system replication. All the required mechanical components were designed and 3D printed with the special care for safety and transferability. A photo of the laboratory setup is displayed in Fig. 1.

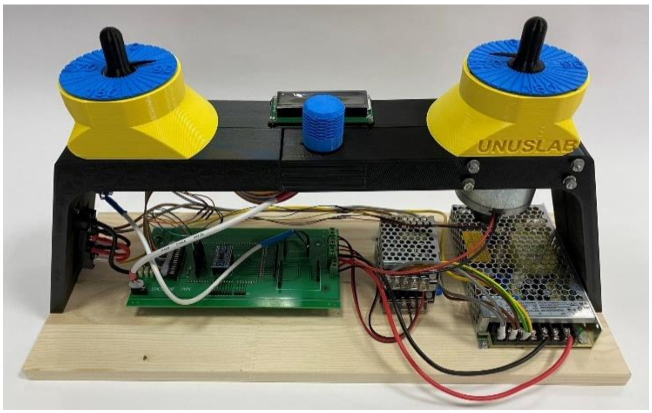

Fig. 1. Velocity servosystem laboratory platform.

A power supply port, an USB port for communication and an USB port for Arduino programming are located in the left-hand side of the platform. The power supplies are integrated into the platform, in the right-hand side, so only a standard PC power cable is needed for powering the system. The laboratory setup was designed to operate both in the stand-alone mode (controlled using the knob and display in the middle) and with usage of a computer. The referent speed/position is selected using the pointer on the left, and the motor rotation is observed using the pointer on the right.

B. Temperature regulation of a thermal process

Temperature regulation system consists of a ceramic resistor with 2.7 Ω resistance and maximum power output of 10 W. It has the role of a control object whose temperature needs to be regulated. A strong MOSFET transistor serves as the actuator of the system, and it allows current switching trough the resistor. The feedback is realized using an analog temperature sensor that is amplified to achieve the highest resolution for the temperature range of 0 ^{0}C – 100 ^{0}C. A photo of the temperature regulation platform is shown in Fig. 2.

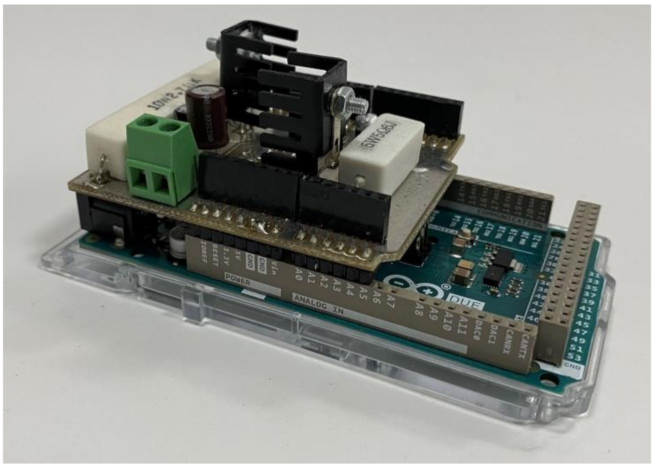

Fig 2. Temperature regulation system

The system also has a secondary ceramic resistor, that has the resistance of 5.6 Ω and maximum power dissipation of 5 W. The resistors represent two different heating elements, that can be utilized either separately or together. This provides flexibility to the platform to be able to emulate different thermal processes.

III. Software architecture

The software architecture consists of three parts: (i) real-time control system implemented on the Arduino DUE, (ii) server that generates the user interface (UI) and (iii) a bridge application that connects the previous two. The system block diagram is given in Fig. 3.

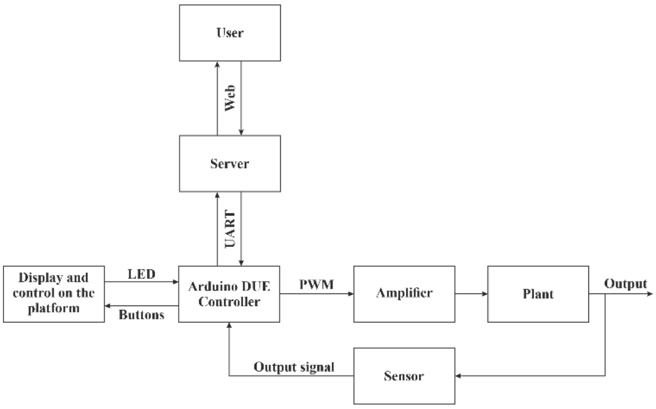

Fig 3. Block diagram of the system.

A. Communication protocols

The real-time control system communicates with the bridge application using universal asynchronous receiver-transmitter (UART) protocol. UART messages are optimized for the minimization of number of bytes, which means that they need to be decoded and passed to the server. This is implemented because of the limited transmission speed of the protocol with really large amount of data (e.g., plot data). The bridge application decodes the UART data and generates Java Script object notation (JSON) messages that are transferred to the server via WebSocket protocol. The communication is organized using a master-slave type of protocol where the server is the master, and the Arduino is the slave. Every message from the server generates a response from the Arduino so any communication error can be easily recognized. If no response is generated retransmission feature has been implemented as well. This provides reliable communication between system components.

B. Real-time control system

The real-time control system consists of three main parts: 1) control loop, 2) communication handler and 3) display handler.

1) Control loop

The control loop must have a real-time component, executed every 4 ms by generating the timer interrupt. It handles the sensor readings for the reference and the output signal, calculate the control signal according to the applied control algorithm, converts that signal from voltage to PWM and forwards it to the selected driver. The control loop can be set to be executed with a longer sampling period, which is the integer product of 4 ms. This enables changing the sampling period of the system. This feature is useful for the experiments that analyze the effects of the sampling period size on a digital control system, as well as for the systems that

 ISBN 978-86-7466-930-3

are considerably slower, such as the temperature regulation system. Several digital control laws can be employed for the control signal calculation. The implemented standard digital controllers are: PD, PID, PID AW, I-P, I-PD and universal controller. PD controller can be used for realization of P controller, by setting the derivative gain to zero. The same goes for PID controller that can be used for the realization of PI controller. PID AW represents the standard PID controller with anti-windup mechanism. I-P and I-PD controllers denote controllers with relocated P and PD actions, respectively. The implemented universal controller can represent any second-order (or less) discrete-time transfer function described by

$$G(z) = \frac{b_2 z^2 + b_1 z + b_0}{a_2 z^2 + a_1 z + a_0} \quad (1)$$

This means that virtually any linear digital controller (up to the second-order) can be tested on the given system.

It is also envisaged that the control signal can be formed using state feedback control approach. In order to implement the state feedback control loop, it is necessary to use the state-space representation of the control plant. The state variables required for the state feedback control can be obtained directly by measuring, or by using state estimation. For this reason, a predictive state observer, a current state observer and a Kalman estimator (filter) were also implemented. The software is realized in a such manner that the real-time simultaneous switching between the controllers can be done. This offers the students the possibility to compare the efficiencies of various controllers.

The desired reference signal can be obtained manually from the user or as the output of the built-in signal generator. The implemented signal generator can generate real-time step, ramp, parabolic, pulse, sine and sawtooth signals. The signal parameters can be arbitrary set and the switching between signals can be done in real-time.

2) Communication handler

The communication handler manages receiving a message, its decoding and generating and sending the corresponding response. The communication handler has various types of error checking, starting from incorrect number of bytes to incorrect data or data types. The number of system parameters that can be changed through the communication handler is 70. This means that every controller gain, every signal generator parameter and every other control loop parameter can be changed by the user through the Web UI. The communication handler is responsible for generating and sending the plot data, which consists of the timestamp, referent signals, output signals, and control signals. The time period of the plot data sending is defined as a user settable parameter, enabling the user to appropriately adjust the plot figure as needed.

3) Display handler

The display handler controls the information presented on the display and the knob that is used for moving through the menus. The display information is useful when performing experiments in the stand-alone mode (not connected to a computer). Still, for the full functionality of the system the Web UI should be used.

C. Web UI

The Web UI is accessible through any Web browser with no special requirements. When granted the access to the platform by the system administrator, the user establishes the connection to the Web UI by typing in the Web address as shown in [6]. Multiple users can access the platform at the same time which enables working in groups, where the members don't need to be in the same location. This has proven to be a useful option, since the pandemic has separated students from each other. The first batch of students that used the laboratory had positive feedback on the look and usability of the Web UI. When accessing the provided Web page, a login screen firstly appears. After entering the correct login information, the control page is shown. The control page is dynamically changing the displayed parameters based on the selected options. Fig. 4. shows the overview of the control page.

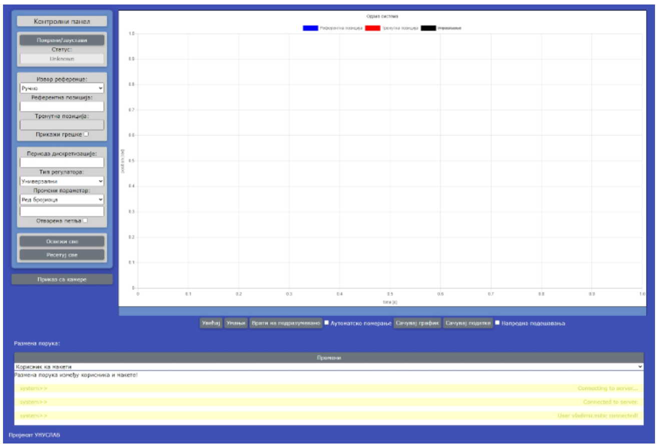

Fig. 4. The control page of the Web UI.

The control page consists of the 1) control panel on the left, 2) control chart on the right and a 3) message panel on the bottom.

1) Control panel

The control panel is used for starting and stopping the system and for system parameter manipulation. It is divided into four sections. The first section consists of the *Start/Stop* button and a system status indicator. The possible system statuses are: Unknown, Stopped, Running and Disconnected. The statuses are described in Table 1.

TABLE I
SYSTEM STATUS DESCRIPTION

Status	Description
Unknown	The bridge application is not responding.
Stopped	Control loop is disabled, and plot data sending is stopped.
Running	Control loop is enabled, and new plot data is being displayed.
Disconnected	Connection with the server is terminated, the page should be refreshed

The second section is intended for reference signal control and output signal display. A dropdown list is used for selecting if the reference signal is entered manually or if it is connected to the signal generator. If a signal generator is selected, the corresponding signal parameters are displayed and can be changed. When the state feedback control is enabled, two input fields are displayed. One for each referent

 ISBN 978-86-7466-930-3

state, and two display fields for each measured state variable. A *Show trend* checkbox displays three new fields, showing the current error, the error derivative and the error integral.

The third section is intended for setting the control loop. The first adjustable parameter is the system sampling period. A dropdown list is used for selecting the loop type and controller type. Loop type can be the standard negative feedback or the state feedback. Controller type is used for selecting one of the implemented digital controllers, explained in the previous section. When selecting the desired controller, corresponding parameters of that controller are displayed and can be tuned. Finally, there is a check box for selecting open or closed control loop. This is mostly useful in system identification exercises.

The fourth section consists of two buttons, one for refreshing of all parameters and one for resetting the all parameters to their default values. Below the control panel, there is a button that is used for displaying the webcam image. It opens a new tab that will display a livestream of the laboratory platform that is obtained via a webcam, which is mounted above the platform.

2) Control chart

The control chart is an interactive chart that can be manipulated in various ways by the user. Signal manipulation can be accomplished on the chart itself. Every signal on the chart can be hidden or shown by clicking on the signal name. Every point on the chart can be displayed in detail by hovering the mouse pointer over it. Additional controls for manipulating the chart are located directly below it. They consist of the following buttons: *Zoom in*, *Zoom out*, *Reset view* buttons and *Auto-scroll* checkbox for closer observations of the response, *Save graph* and *Save data* buttons for storing the response and an *Advanced settings* checkbox. *Save graph* button stores the presented response as an "Graph.png" file while the *Save data* button stores the response in a "data.csv" file. Every row in the file represents one timeseries, and every column one signal. This makes the data easily readable, and importable into various software packages (like Matlab) for further detailed analysis. The *Advanced settings* checkbox displays three new input fields: *Plot acquisition period*, *Number of seconds to store* and *Number of samples to store*. *Plot acquisition period* defines the period at which the platform sends the data to the user. This period can be different to the control sampling period if we do not need so much data collected and displayed. *Number of seconds to store* defines the timespan that is shown on the chart and that will be stored with the save button. *Number of samples to store* represents the number of timeseries points that is displayed and stored. Because these two fields affect the same property (timespan of the plot), only one of them should be changed.

With this kind of interactive control chart, it is obvious that it alone is sufficient for any standard system analysis, which is done when executing a laboratory exercise. This enables the students to observe the experiment, notify and immediately correct any unexpected system behavior.

3) Messaging system

The messaging system is a feature that is designed for student groups that are scattered in different places. Using the toggle button on the top of the messaging panel, different types of messaging channels can be selected. Firstly, there is the *User-device* channel that is used for verifying that all the parameters are set correctly. Secondly, there is the *User-user* channel which is organized in a standard chat form and enables students to collaborate more effectively while performing the given task.

IV. Laboratory Exercises

The realized laboratory platforms have a lot of teaching potential for the students, and they can provide practical engagement for students to better learn and understand control theory. First, students can learn about system identification techniques for different types of control plants. For example, students can record the step response of the system by removing the feedback loop and setting the digital controller to PD with the gains $k_p = 1$, $k_d = 0$. In this way, the value of the referent signal is directly passed to the plant. They can save the response data and use Matlab for system identification. By performing this on the servosystems, they learn about first- and second-order transfer function identification. In case of the thermal plant, they can learn about identifying a system with transport delay.

The second set of experiments are devoted to time-discretization process and signal reconstruction in digital control systems. Students can observe the effects of the sampling period selection on a system performance. They can experimentally determine critical gain and critical frequency when system reaches stability margin, Fig. 5. Also, they can calculate steady state errors to different input signals and experimentally validate them.

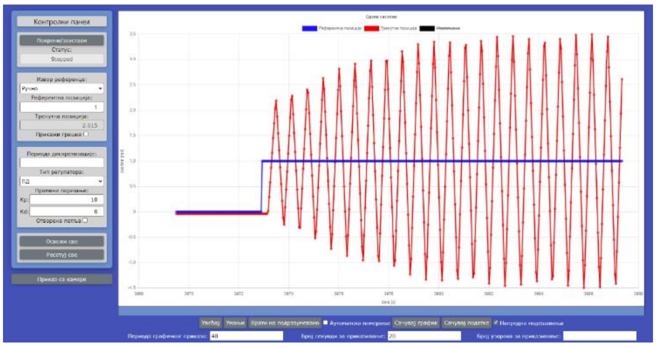

Fig 5. Experimentally determining the critical gain and frequency of a closed loop system.

When it comes to digital control systems design, the implemented digital controllers give the students a lot of possibilities. Starting from standard control loops, they can design P, PI, PD, PID controllers, lead-lag compensators and compare the system responses that they provide. Also, the students can experience some issues related to practical implementation of a control system. One of them is the saturation of the applied control signal, which can greatly affect the performance of a system. Consequently, possibilities are envisaged to implement digital controllers with relocated P and D actions that can reduce control signal peaks, as well as to enhance a control system with an anti-windup structure. The latter is crucial for temperature regulation system because of the large integrator windup produced by the dynamic properties of the system as well as the present transport delay, Fig. 6.

The state feedback control structure implementation widens even more the learning possibilities of the platform.

 ISBN 978-86-7466-930-3

By designing the state feedback controller, students get a better understanding of the state-space representation of a system and its practical usage. The implemented prediction state observer, current state observer and Kalman estimator, as a special type of current state observer, offers possibilities to students to try their design and get insight into state estimation and its possible usages. An example of application of current state observer in state feedback control is given in Fig. 7.

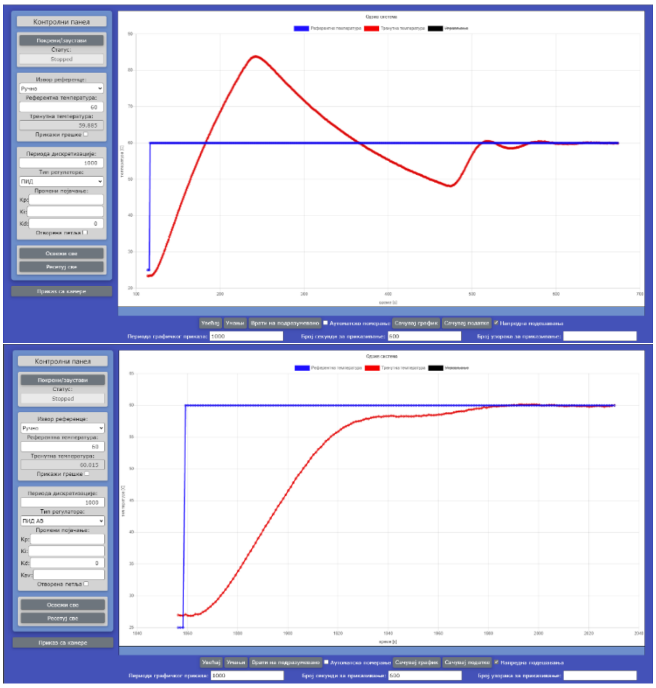

Fig. 6. Temperature regulation system without anti-windup structure (top), and with it (bottom).

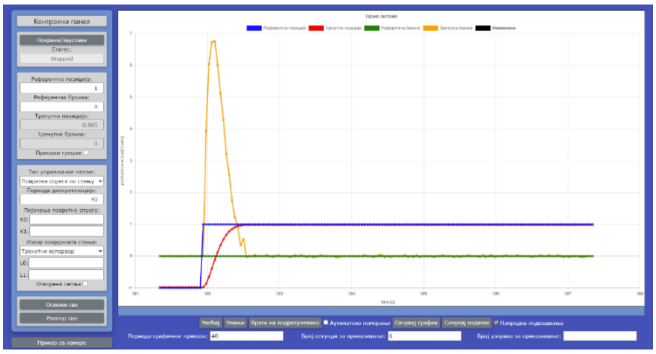

Figure 7. State feedback control implemented with current state observer.

V. Conclusion

The designed laboratory platform represents a universal solution for teaching of different types of digital control systems. User friendly and modern UI makes it very easily understandable for students. Its inexpensiveness makes it a viable option for a lot of educational facilities. The applied motors and drivers can be easily replaced with any motor power and driver combination, without losing any of the functionality of the exercises. The Arduino microcontroller used here enables for realization of even more complex control loops, which can be useful to students at a higher educational level. The scalability of the system makes it work with any digital system with UART protocol and this type of communications handler. The special value of this platform is the possibility of remote access and performing laboratory exercises from a distant location, which was especially useful for students in a pandemic. Future work will be directed to introduction a cooling element to the temperature regulation system, implementing force feedback loops as well as a cascaded control system with the positional outer loop and velocity inner loop for jerk reduction. This will broaden the possible systems for which students can design and test different control structures.

Acknowledgment

Part of this research was financed by the Serbian Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] M.B. Naumović, D. Živanović: "Remote Experiments in Control Engineering Education Laboratory", International Journal of Online Engineering, ISSN: 1861-2121, vol. 4, No. 2, 2008, pp. 48-53.

[2] I.Perenc, T. Jaworski, P. Duch, "Teaching programming using dedicated Arduino Educational Board", Computer Applications in Engineering Education, Vol. 27. Issue 4, July 2019, pp 943-954.

[3] F.A. Candelas, G.J. García, et. al., "Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics", IFAC-PapersOnLine, Vol 48, Issue 29, 2015, pp 105-110.

[4] J Kinchin, "Using an Arduino in physics teaching for beginners", Physics Education, Vol 53, No 6, 2018

[5] AEAT-6010/6012 Magnetic Encoder, https://docs.broadcom.com/docs/AV02-0188EN

[6] http://25.49.3.248:8080/WebPSAU_Position/

 ISBN 978-86-7466-930-3