Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-27 (2019) 217–221

Arduino Support for Personalized Learning of Control Theory Basics ⋆

Pavol Bisták *

* *Institute of Automotive Mechatronics, University of Technology in Bratislava, Slovakia (e-mail: pavol.bistak@stuba.sk).*

Abstract: This paper discusses a possibility how to make learning of control theory basics more interesting for an individual learner. Specifically, an approach of learning by doing is applied when using a low-cost Arduino architecture for each student. As an example of dynamical systems simple RLC-circuits have been chosen. The advantage of Matlab/Simulink support of Arduino platform has enabled students to perform programming tasks from known environment. This paper concentrates on a limited part of the control theory, namely the static and dynamic characteristics. Particularly, in this paper we will present how to construct different RLC circuits, measure following characteristics: static I/O characteristic, step, impulse and frequency responses and identify their parameters.

Keywords: control theory, learning by doing, RLC circuits, static and dynamical characteristics, microcontroller Arduino UNO

1. INTRODUCTION

Control theory belongs to exciting fields of study but many students see it like continuation of mathematics with a lot of formulas and derivations. In fact, it is a kind of applied mathematics but in order to attract the students we have to present the control theory in the way they will see its sense in practical applications from the very beginning, especially when the technological development offers an environment for low-cost solutions.

It was not always true several decades ago. This was the time when computers were not accessible and the students had to calculate everything by hand. Experiments could have been performed with an expensive measurement apparatus, usually for big groups of students. Later, the period of personal computers allowed to individualize learning. With the simulation software releases (e.g. Matlab, Scilab, Modelica, Octave) the students could at least verify results calculated by hand. The prices of experiments were decreasing but still not enough to be at disposal for each student during his/her exercises. The experimentation part of study has been enabled for work in small groups or sharing the equipment for a limited time. The different progressive method has been provided by building remote laboratories when the students have accessed the real equipment via the Internet (Sáenz et al., 2015; Kalúz et al., 2015; Bosak and Zakova, 2015).

Nowadays the cost of simple laboratory equipment for demonstration of basic principles of control theory decreased rapidly thanks to the implementation of microcontrollers. This kind of equipment (e.g. thermo-opto-mechanical system, hydraulic system, magnetic levitation system, etc.) uses the USB interface to be connected directly to the students' laptops without necessity to buy an expensive measurement card (Huba and Huba, 2015; Huba and Halás, 2011; Malatinec et al., 2014; Takács et al., 2019). There exist also microcontroller based open source hardware solutions that can be downloaded free by any institution and used for low-cost production of desired amount of the laboratory equipment (Takács, 2019).

The approach in learning of control theory presented in this paper is a little bit different. It also relies on technological development in the field of microcomputers but it offers a possibility to every student to build his/her own simple laboratory equipment represented by a basic electrical circuit and connected through the microcontroller to his/her laptop. In education this approach is usually classified as learning by doing.

This paper concentrates on a limited part of the control theory, namely the static and dynamic characteristics. The microcontroller Arduino Uno has been chosen because of its widespread adoption, Matlab and community support, and of course, its price. In literature one can find several expamples when the microcontroller Arduino Uno has been used for similar purposes (Suwondo and Sulisworo, 2017; Weis, 2014; Messner et al., 2017). Particularly, in this paper we will present learning by doing methodology how to construct different RLC circuits and measure following characteristics: static I/O characteristic, step, impulse and frequency responses. We also discuss advanteges and limitations of such an approach from the pedagogical view.

The content of the paper is structured as follows. After the Introduction chapter the second chapter describes the microcontroller in combination with RC circuit and its possible usage within the Matlab/Simulink environment. The next chapter presents the static and dynamic responses. The Arduino and Matlab code are included as well as graphical outputs of measurements. In Conclusion

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

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.759

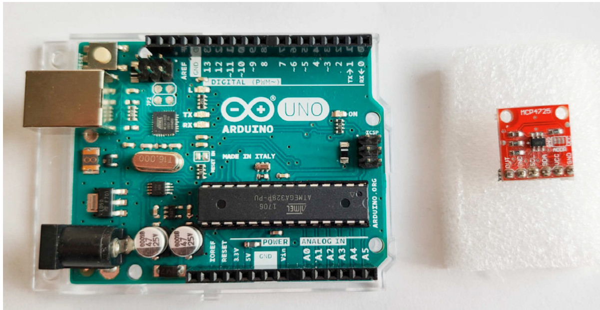

Fig. 1. Arduino Uno board and digital-analog converter MCP4725

chapter the results are evaluated and further development / modification possibilities are discussed.

2. ARDUINO UNO - LOW-COST MICROCONTROLLER PLATFORM

This paper is oriented to teaching the control theory basics using learning by doing methodology. The choice of a low-cost microcontroller platform corresponds to this purpose. Although there are several possibilities (e.g. STM, TI) the choice of Arduino Uno has been obvious when taking into account its popularity, support and price. Arduino Uno R3 uses the 8-bit microcontroller ATmega328P that is placed on the development board equipped with input/output (I/O) pins, connectors (USB, power) and other necessary components for operation of microcontroller (Fig. 1). For our purposes it is important that Arduino Uno has analog inputs (6) for signal measurements and digital I/O pins (14) from which 6 pins support PWM signals. Arduino Uno does not have any analog outputs therefore to produce wave signals for frequency characteristics we will have to use an additional board that will serve as a digital-analog converter (12-bit MCP4725). The board of Arduino can be powered also by the USB cable.

From the family of RLC-circuits as an introductory dynamical system the RC-circuit has been chosen. It consists just from two passive electrical components, i.e. the resistor and the capacitor as depicted in the Fig. 2. Even students that have no practical experiments with building electrical circuits should be able to construct such an easy circuit by connecting these basic elements with wires on a breadboard. The physical connections of the RC-circuit with the Arduino Uno microcontroller can be seen from the Fig. 3. One can be confused of so many wires but this is caused as a consequence of using D/A converter. Even this converter is effectively activated through I2C it still has six ports to be connected. Basically, the RC circuit is powered from the D9 pin of Arduino. Only in the case of frequency responses the RC circuit is alternatively powered from the output of the D/A converter. The output of the system represented by the voltage on the capacitor is measured using the A0 pin of the Arduino that serves as an 10-bit analog-digital converter. The measurements were carried out with the following numerical values of the electrical components: $R = 4700\Omega$, $C = 3,3\mu F$.

To program the Aruduino boards we can use several tools from which the Arduino Integrated Development Environment (IDE) is the most known (Fig. 6). This

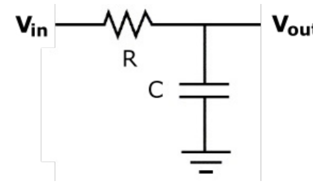

Fig. 2. RC circuit

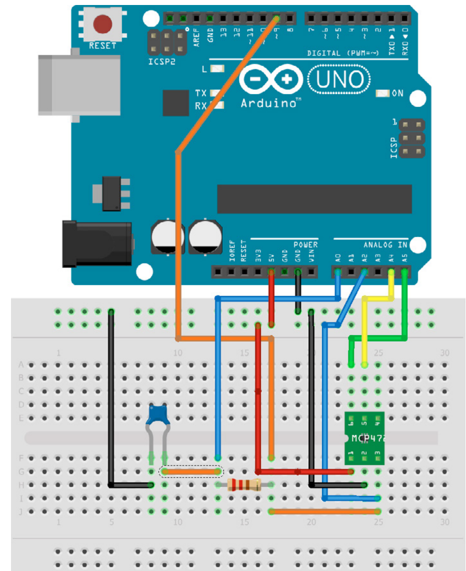

Fig. 3. RC circuit with Arduino and D/A converter in Fritzing

enables to write the code for Arduino in the language very similar to the C++ programming language. But programming language for Arduino is not limited to C++. You can use well-known languages as Python, Java, C# and also the Matlab/Simulink in which we are the most interested because this is the language for most of the control theory courses. In fact, we will combine C++ programming in Arduino IDE with Matlab programming. Arduino will send measured data to the Matlab where this will be visualized and identified.

3. MEASUREMENT OF BASIC CHARACTERISTICS OF DYNAMICAL SYSTEMS

After the students have built the simple real RC circuit, connect it to the Arduino Uno board (4) and install all necessary software (Arduino IDE and Matlab) they can proceed with the measurement of basic characteristics of such a system. In the following sections we will describe how Arduino can be used for measurement of input-output characteristic, step and frequency responses.

3.1 Input-output characteristic of RC circuit

Although the input-output characteristic does not provide us with the information about the dynamics of the circuit it still belongs to the most important information in the control theory. It allow us to identify the gain of the system as a ratio of steady-state output to the constant input. By measurement at different inputs we can identify if the system is linear or nonlinear one. The gain parameter and the linearity of the system play crucial role in the control

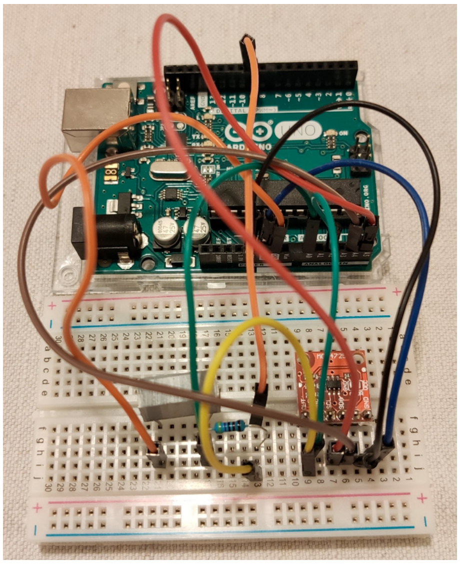

Fig. 4. Real RC circuit connected to Arduino and D/A converter

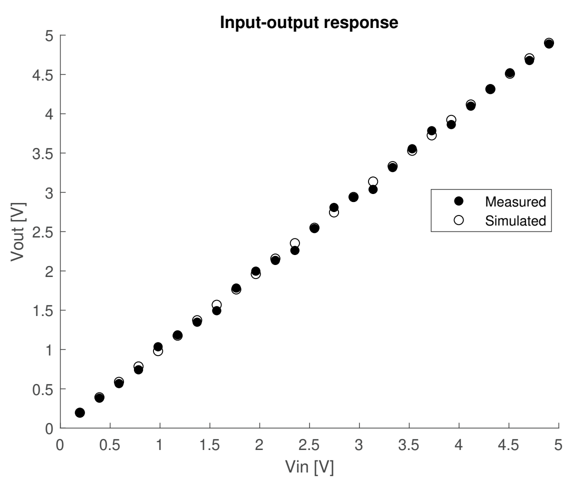

Fig. 5. Input-output characteristics of RC circuit

theory and therefore it is important to let students work with a real systems from the very beginning.

The RC circuit is the simple one and therefore it is suitable for beginner in the control theory. In the Fig. 5 they can see the linear input-output characteristic corresponding to the RC circuit. From this example students can not only identify the linearity of the system but calculate the gain of the system as well when they express the quotient of the linear relation. It determines the slope of the I/O characteristic also visible from the plot in the Fig. 5.

The question is what students have to do in order to be able to measure such characteristics. The must write to short programs. One for Arduino and another one for Matlab. At this stage they should already know the basics of the C-language and therefore it could not be difficult for them to understand or modify the code that is displayed in the Fig. 6. They should also be familiar with Matlab to have possibility to customize the code presented in the Fig. 7. Both software use serial communication. In fact, Matlab waits until it starts receiving message. Arduino measures the real data and sends it to the Matlab that

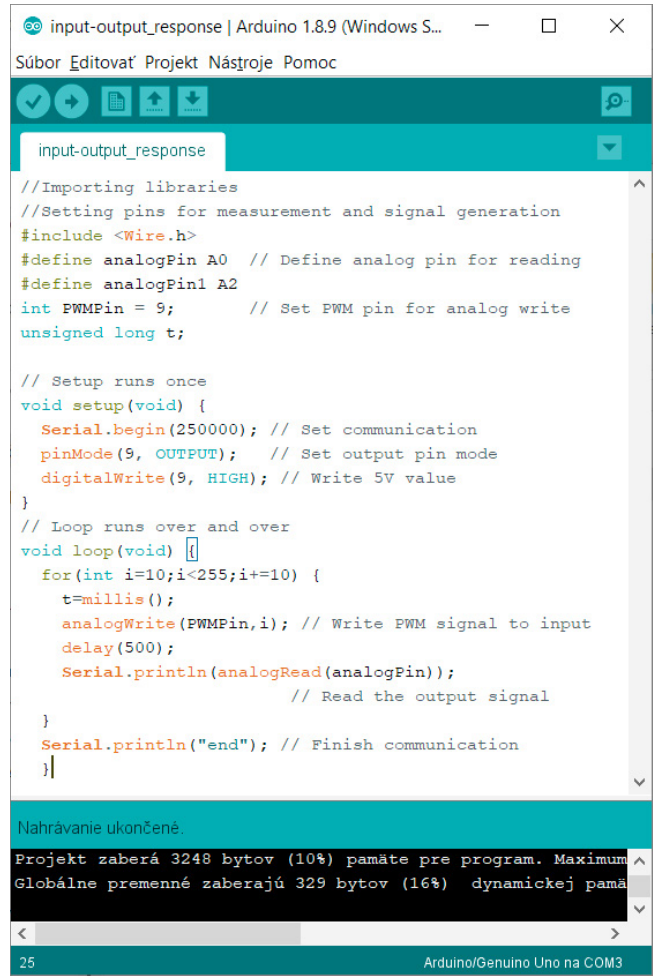

Fig. 6. Code from Arduino IDE for performing input-output characteristics

after collection all data starts to plot them. The resulting I/O characteristic is depicted in the Fig. 5 where it is visible that in this case there is no big difference between the measured and simulated characteristics.

3.2 Step response of RC circuit

After knowing the gain of the system, students would like to investigate the dynamics of the RC circuit. The step response is the first dynamic characteristic that shows the behavior in relation with time. In this simple example this corresponds to the identification of the time constant of the RC circuit. From the theory we know that the time constant of the first order dynamical system can be calculated using step response from the time when the output reaches the 63% of the steady-state value.

To program the measurement of the step response there is no big difference in comparison with the measurement of I/O characteristic. In comparision with the previous case now Arduino sends to Matlab two values: output and time. We do not present the Arduino and Matlab codes here. But the resulting step response can be found in the Fig. 8. As in the I/O characteristic also here the measured and simulated step response do not differ significantly. Of course, one can mention that the real data are influenced by a small noise. After finishing this experiment students

Input-output characteristic

```
instrreset
close all;
% Setting Arduino port for serial commmunication
s=serial('COM3','BaudRate',250000);
fopen(s);
Vin=10:10:250;  % Generating input values
Vout=[];        % Array for output values
t0 = clock;
while (etime(clock, t0) < 20 )
    str=fscanf(s);  % Reading messages
                    % from serial communication
    if (strncmp(str,'end',2))
        break
    else
        hodnoty = textscan(str,'%f');
        % Conversion of read values
        Vout(end+1)=cell2mat(hodnoty(1,1));
    end
end
% Conversion from digits to Volts
for i=1:length(Vout)
    Vin(i)=Vin(i)*(5/255);
    Vout(i)=Vout(i)*(5/1024);
end
% Plots
figure
scatter(Vin,Vout)
xlabel("Vin [V]")
ylabel("Vout [V]")
title("Input-output response")
fprintf('end')
instrreset
```

Fig. 7. Matlab code for performing input-output characteristics

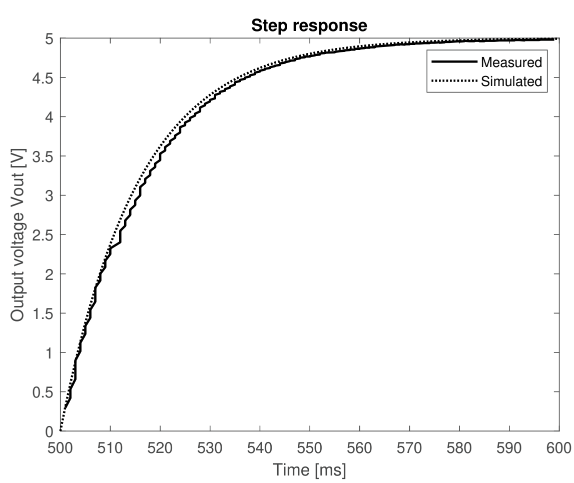

Fig. 8. Step responses of RC circuit

should know the gain of the system and also its time constant. So they can continue with advance topics. They can start to design simple controllers or they can continue with measurement of frequency responses.

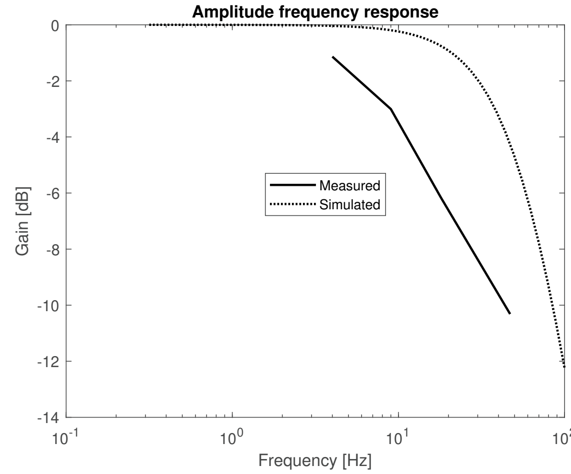

Fig. 9. Amplitude frequency responses of RC circuit

3.3 Frequency response of RC circuit

To measure the frequency characteristics with the only help of Arduino Uno is not so easy task as the previous ones. Here the limits of Arduino as a low-cost platform are apparent. The main problem is caused by the lack of a harmonic signal generator. Arduino itself is not able to produce sinusoidal signal with higher frequency. Although we extended the Arduino with D/A converter and the sinusoidal values have been pre-calculated and stored in tables we were able to measure frequency responses up to 50 Hz. The measurement was not precise and this time the differences between the measured and simulated response are significant (Fig. 9).

Although this activity could be for students the most difficult one in comparison with the previous cases we recommend to let student measure also the frequency responses at least in a few points because this topic of the control theory study is usually not well understood.

4. CONCLUSIONS

In order to provide each student with the equipment for experimentation, in this paper we focused to the usage of the Arduino Uno platform for the measurement of static and dynamic characteristics of dynamical systems. Arduino is able to communicate through the serial interface with Matlab that serves for comfortable visualization and archiving of the measured data. The students are engaged in the learning process when they have to build their own circuits and program microcontrollers to measure the characteristics. Their work could be personalized because Arduino low-costs platform is accessible for everyone. We have presented precise measurement in the case of I/O and step characteristics but we have also mentioned the disadvantages of the low-cost platform when measuring the frequency responses. For the future we will try not to replace the Arduino by expensive signal generators and scopes but find cheap solutions using electronic oscilators and a little bit powerful types of microcontrollers.

REFERENCES

T. Bosak and K. Zakova. Node.js based remote control of thermo-optical plant. *Proceedings of 2015 12th Interna-*

tional Conference on Remote Engineering and Virtual Instrumentation, REV 2015, pages 209–213, 2015.

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

T. Huba and M. Huba. Experimenting with a nonlinear thermo-opto-mechanical system TOM1A. *IFAC-PapersOnLine*, 48(29):188–193, 2015. doi: 10.1016/j.ifacol.2015.11.235.

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.

T. Malatinec, V. Popelka, M. Huba, and P. Hudačko. Laboratory model helicopter control using a lowcost arduino hardware. *Proceedings of the 2014 15th International Carpathian Control Conference, ICCC 2014*, pages 326–331, 2014. doi: 10.1109/CarpathianCC.2014.6843621.

B. Messner, R. Hill, and JD. Taylor. Control Tutorials for Matlab and Simulink - Activity 2 Part (a): Time-Response of an Inductor–Resistor–Capacitor (LRC) Circuit. `http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_LRCcircuitA`, 2017. Accessed: 2019-5-23.

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.

N. Suwondo and D. Sulisworo. Hands-on Learning Activity Using an Apparatus for Transient Phenomena in RC Circuit Based on Arduino UNO R3-LINX-Labview. *iJOE International Journal of Online and Biomedical Engineering*, 13(1):116–124, 2017. doi: 10.3991/ijoe.v13i01.6317.

G. Takács. AutomationShield - Control Systems Engineering Education. `www.automationshield.com`, 2019. Accessed: 2019-09-24.

G. Takács, T. Konkoly, and M. Gulan. Optoshield: A low-cost tool for control and mechatronics education. *2019 12th Asian Control Conference, ASCC 2019*, pages 1001–1006, 2019.

S. Weis. Using an arduino to measure frequency response and current-voltage device characteristics in electronics labs. *Proceedings of the 2014 ASEE Gulf-Southwest Conference, New Orleans, Louisiana, USA*, 2014.