



This is a repository copy of *Introducing digital controllers to undergraduate students using the TCLab Arduino kit.*

White Rose Research Online URL for this paper:
<https://eprints.whiterose.ac.uk/157786/>

Version: Published Version

Proceedings Paper:

de Moura Oliveira, P.B., Hedengren, J.D. and Rossiter, J.A. orcid.org/0000-0002-1336-0633 (2021) Introducing digital controllers to undergraduate students using the TCLab Arduino kit. In: IFAC-PapersOnLine. 21st IFAC World Congress 2020, 11-17 Jul 2020, Berlin, Germany. Elsevier , pp. 17524-17529.

<https://doi.org/10.1016/j.ifacol.2020.12.2662>

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Introducing Digital Controllers to Undergraduate Students using the TCLab Arduino Kit

P. B. de Moura Oliveira*, John D. Hedengren ** and J. A. Rossiter ***

* INESC-TEC, Department of Engineering, University of Trás-os-Montes and Alto Douro (UTAD), Vila Real, Portugal

** Department of Chemical Engineering, Brigham Young University, Provo, UT, USA

*** Department of Automatic Control and Systems Engineering, University of Sheffield, S1 3JD, UK

(e-mail: oliveira@utad.pt; john.hedengren@byu.edu; j.a.rossiter@sheffield.ac.uk)

Abstract: Many undergraduate engineering students have just a single introductory feedback control course in their study list. Often the curricula found in such courses is totally based on continuous time-domain classic control techniques. However, currently most control design techniques are implemented in digital machines. Thus, digital control concepts should be covered in introductory control courses. In this paper, the issue of how to implement and test digital industrial controllers is addressed. Teaching experiments based on the APMonitor temperature control lab (TCLab) are proposed. It will be shown that TCLab as an Arduino based portable kit, provides an excellent means to test digital controllers, as it is a compact and portable lab to be used by lecturers and students. While there are many low-cost and portable hardware options for teaching dynamics and control, a novel aspect of this paper is the digital control education methods that are validated with classroom experience, particularly with Biomedical and Bioengineering students. Preliminary results are presented.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0>)

Keywords: On-off Control, PID Control, Control Engineering Education, Arduino.

1. INTRODUCTION

Many first-degree engineering programs (3 years) curricula include a single feedback control course. Depending on the type of engineering and the master's degree selected by students, they can have (or not) more specialized control courses (e.g. digital control, system identification, process control, etc.). Some engineering students are more likely not to have second control engineering courses than others (e.g. Aeronautical engineering). This is the case for both the UTAD first degrees on Biomedical Engineering and Bioengineering. Recently this issue was under debate by the IFAC control education community (e.g. PID-18, 2019). Until such a standard curriculum for this type of courses emerges, lecturers face the dilemma of what subjects to teach. The great majority of introductory feedback and process control reference books (e.g. Dorf and Bishop, 2017; Seborg *et al.*, 2016) are mostly based on classical continuous time and frequency domain techniques. However, currently in practical applications most feedback control techniques are implemented in digital devices (ranging from computers, programmable logic controllers, digital signal processors, etc.). It makes sense (from the authors' point of view) to introduce digital control content to undergraduate engineering students. How far can one go in this digital control introduction is another issue, which depends also on students' prior acquired skills (e.g. Z-transforms taught in digital signal processing courses). The experimental digital material reported here is restricted to the main industrial control modes of implementation, namely: on-off, proportional action, integral action and derivative action.

A crucial issue regarding control engineering is testing and validation of theoretical concepts in practical classes. Usually this is carried on using kits which replicate at laboratory scale real

systems (e.g. Feedback Instruments, 2019; Quanser, 2019). This type of educational equipment has two major drawbacks: *i*) they are quite expensive, and thus not affordable by many educational institutions and *ii*) they require physical access to laboratories by students. Out of the class time, students' access to laboratories may be difficult for several reasons (scheduling of free slots, supervision of technical staff in certain hours, etc.). While for some types of processes and in some institutions, access to laboratories can be performed remotely using the Internet (e.g. Fabregas *et al.* 2011), this may also require significant investments. For these reasons, the development of affordable control kits which can be purchased and used by everybody (students and teachers) is a highly desirable feature. Even better, if these kits are small sized and portable. These devices can be termed take-home laboratory kit (Rossiter *et al.*, 2019). For temperature control, the Arduino based Temperature Control Lab (TCLab) proposed by (Hedengren, 2019; Hedengren *et al.* 2019) fulfills all the prior requirements.

There are several reports of using an Arduino (e.g. Irigoyen *et al.*, 2013; Reguera *et al.*, 2015; Prima *et al.*, 2017) for performing feedback control experiments. As it is shown hereafter, three factors distinguish TCLab from other Arduino based kits: *i*) it is fully assembled and ready to be used; *ii*) the software to use TCLab is freely available both for Python and MATLAB/Simulink (Hedengren, 2019); *iii*) supporting educational videos and material is freely made available in (Hedengren, 2019; Kantor, 2019).

Despite the development of more advanced control techniques, there are many industrial loops controlled by proportional, integrative and derivative (PID) controllers. The practical relevance of this type of control have motivated the development of a wide range of design techniques (e.g. Åström and Hägglund, 2006;

Visioli and Vilanova, 2012). This relevance makes PID control a lectured topic in most introductory control courses (Silverstein *et al.*, 2016). How to implement a PID control digitally is a relevant issue also addressed in this work. PID tuning techniques can be based on open-loop and closed-loop procedures. Following the same approach used in this experience first report (Moura Oliveira and Hedengren, 2019), the identification of a first order plus time delay (FOPTD) model from an open-loop step response is performed using a classical technique: the two-point method (Sundaresan and Krishnaswamy, 1978) and a nature inspired optimization technique: the particle swarm optimization (PSO) technique (Kennedy and Eberhart, 1995). PSO has been used successfully in control engineering education (Moura Oliveira, 2018) and PID design (Freire *et al.*, 2017). The controller tuning is performed using the well-known Cohen and Coon (CC) (1953) rules, which were previously shown to be adequate for this type of system (Moura Oliveira and Hedengren, 2019). Note, that more recent tuning techniques (e.g. Skogestad, 2003), are taught to the students performing this experiment and can be used for comparison.

The rest of the paper is organized as follows: Section 2 presents TCLab basics. Section 3 describes TCLab student's familiarization. Section 4 presents system model identification; Section 5 addresses controller design and students' perceptions; Finally, section 6 concludes the paper.

2. APMONITOR TEMPERATURE CONTROL LABORATORY BASICS

The TCLab is an Arduino based temperature control kit based on the hardware shown in Figure 1, it requires a USB connection to a computer and electric power supply.

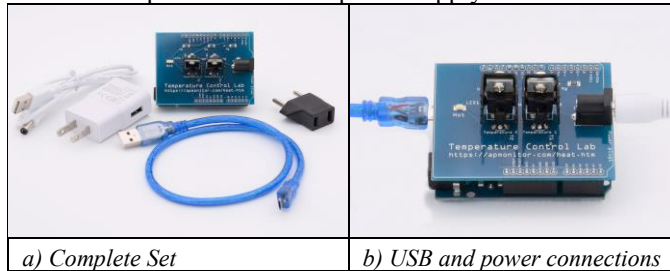


Fig. 1 TCLab Temperature Control Lab components and connections.

The TCLab has two temperature sensors (TMP36) and two actuator heaters (TIP31C NPN Bipolar Junction Transistors in package form TO-220) (see Figure 2).

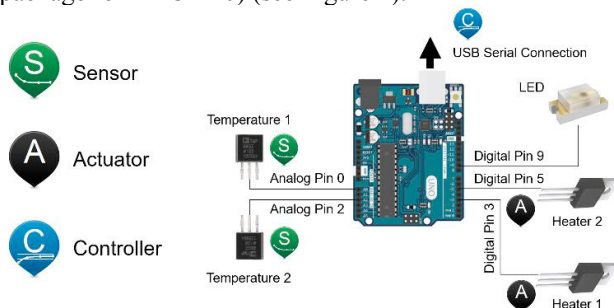


Fig. 2 APMonitor Temperature Control Lab (TCLab) kit schematic.

The TMP36 sensors ($T1$ and $T2$) send temperature readings to the Arduino using a voltage signal (mV). These voltage readings (ranging between 0 and 3300 mV) are digitalized in the Arduino with a 10-bit Analog to Digital Converter (ADC) resulting in 1024 discrete levels (DL). This is converted to millivolts with $mV = \frac{3300}{1024} DL$ and to temperature with $T = 0.1 mV - 50$. TMP36 sensor accuracy is $\pm 1^\circ C$ at $25^\circ C$ and a maximum of $\pm 2^\circ C$ throughout the entire sensing range of $-40^\circ C$ to $150^\circ C$. If the temperature sensor exceeds $100^\circ C$, the Arduino pre-loaded firmware turns the heaters off to protect the equipment and the users. The TCLab has two heaters ($Q1$ and $Q2$), which are transistor heaters (TIP31C). This type of BJT transistor is not commonly used as heaters but can expel heat when used in power, audio, and ON/OFF switch applications. Because the TIP31C is less efficient than a MOSFET power transistor, it acts as a heater and switch. The TIP31C heater is controlled through Pulse Width Modulation (PWM). PWM is one method to emulate an analog output with a digital (ON/OFF) switch. The Arduino cycles from 0 to 100% in rapid succession (500 Hz or every 2 milliseconds). The digital output is 8-bits with 256 discrete levels represented as integers between 0 and 255. The transistor heater can be set between 0 and 100% output by the Arduino such as $Q1$ to 65% for 10 seconds and $Q2$ to 85% for 5 seconds. The maximum power output of the heater with a 5V power supply is about 1 Watt although that depends on the power supply. Some of the power from the supply is dissipated over the power cable wire that has a thickness of 20 American Wire Gauge (AWG). The power cable connects the USB power source (either a computer port or the power supply) to the TCLab shield to provide a source for the heat generation. The heaters are set to different power levels in the Arduino firmware. This is by design to make the gain for heater 1 and heater 2 different. When heater 1 is set to 100%, a signal of 200 out of 255 is sent to the Arduino. When heater 2 is set to 100%, a signal of 100 out of 255 is sent to the Arduino for half the power of heater 1. This difference in output gives a process gain difference by a factor of 2. More information regarding TCLab can be found in (Park *et al.*, 2020).

The reported experiment is organized in three main parts: 1) Introducing TCLab to students; 2) System Model Identification; 3) Controller design. Part 1 is described in section 3, while the other two in subsequent sections.

3. TCLAB STUDENTS FAMILIARIZATION

This section outlines the TCLab programming framework. TCLab can be programmed both using Python and a MATLAB/Simulink environment, and some introductory programs and models are freely provided in (Hedengren, 2019). The first decision to make before using TCLab in classes is whether to use Python and/or MATLAB/Simulink? Note that to use TCLab with MATLAB/Simulink it is necessary to install the respective Arduino support libraries. In this work, a mixed approach is adopted: Python programs are used to perform all the identification and control experiments. As UTAD students use MATLAB/Simulink in previous course classes, the test TCLab data analysis for model parameter estimation, controller design and graphics are performed with MATLAB/Simulink. From the Python introductory programs available (Hedengren, 2019) the

following TCLab test sequence was used: *i*) Testing the TCLab LED (*test_LED.py*, see Figure 3 a)) by gradually decreasing the LED power. As it is observed from line 5 of the code in Figure 3a), all the TCLab programs require another program (*tclab.py* or ‘pip install tclab’) which establishes and closes the Arduino connection. *ii*) Testing the TCLab Temperature Sensors and Actuators (*test_Temperature.py*): reads the thermistors temperatures (T_1 and T_2) and turns on heaters (Q_1 and Q_2); *iii*) Testing the TCLab Heaters (*test_Heaters.py*) and obtain an open-loop step response actuating just the heater Q_1 . This test result is represented in Figure 3b).

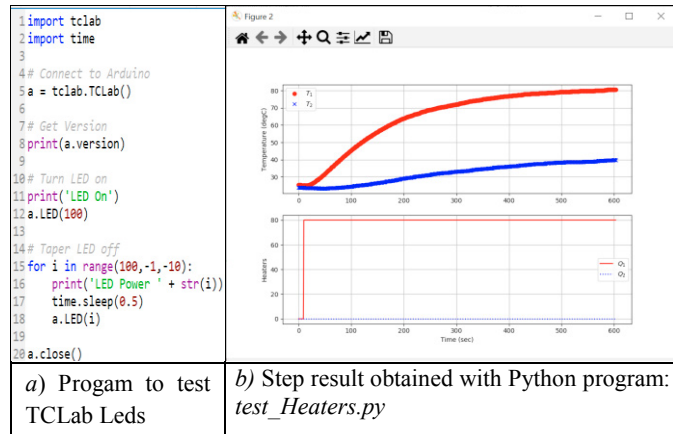


Fig. 3 TCLab Temperature Control Lab components and connections.

4. SYSTEM MODEL IDENTIFICATION

In Figure 4, a flowchart describing the PID controller design procedure is presented. The students perform model identification by estimating the FOPTD parameters from the open-loop step response obtained from the TCLab. Similar to what was reported in (Moura Oliveira and Hedengren, 2019), two approaches are adopted: *i*) a classical technique based on the two-point method ($2pt$) (Sundaresan and Krishnaswamy, 1978) and *ii*) an optimization technique based on the PSO. The $2pt$ FOPTD can be evaluated as follows:

$$K = \frac{y_{ss}-y_0}{u_{ss}-u_0} = 0.76 \quad (1)$$

$$T = 0.67(t_{85}-t_{35}) = 137s \quad (2)$$

$$L = 1.3t_{35} - 0.29t_{85} = 28.6s \quad (3)$$

with: y_{ss} , u_{ss} , y_0 and u_0 representing the system and controller outputs steady-state and initial values; t_{35} and t_{85} , the time for the first (35.2%) and second point (85.3%), K , T and L the dc gain, time constant and time-delay, respectively. Thus, the $2pt$ model obtained for the advanced TCLab kit is represented by:

$$G_{2pt}(s) = \frac{0.76}{1+137s} e^{-28.6s} \quad (4)$$

The PSO algorithm for the FOPTD identification has a swarm size 30 particles, characterized by a position, x , and velocity, v , vectors. The particle position is represented by a tri-dimensional vector representing the FOPTD parameters $[K, L, T]$. The swarm positions are initialized randomly in the search space within the following variable boundaries $[0.1,1]$, $[30 \ 170]$, $[5 \ 50]$, for K , T and L , respectively. Once the $2pt$ model is identified, those initial values are used to select the model parameter boundaries to perform the PSO optimization. While a PSO script was

provided to students, which allows them to change the PSO settings, Matlab® has available a specific PSO function *particleswarm* which can also be used. PSO minimizes the integral square error (ISE) between the model output (simulated using a Simulink model) and the response obtained from the TCLab open-loop response.

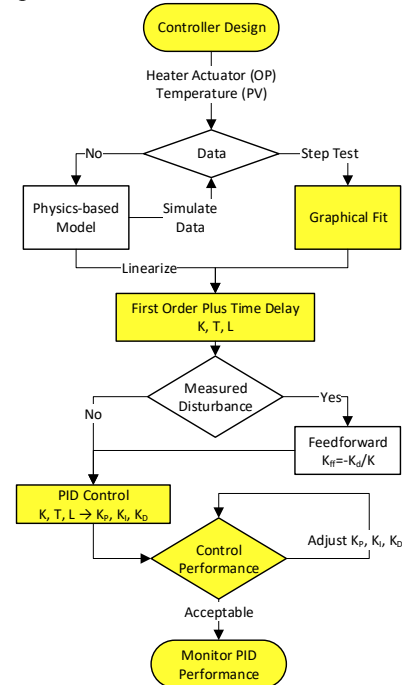


Fig. 4. Flowchart for the Identification and PID Control approach with TCLab in the reported experiment.

Each swarm particle, i , velocity and position are updated every iteration, t , using (5) and (6), respectively:

$$v_i(t+1) = \omega v_i(t) + c_1 \phi_1 (b_i(t) - x_i(t)) + c_2 \phi_2 (g(t) - x_i(t)) \quad (5)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

with the following representation: c_1 and c_2 are the cognitive and social constants (here $c_1=c_2=2$); ϕ_1 and ϕ_2 are randomly generated numbers within $[0,1]$; b is the best i -particle individual position obtained until iteration t ; g is the global swarm best position (fully connected neighborhood topology) and ω is the inertia weight decreased in the interval $[0.9 \ 0.4]$ along the search (100 iterations). A maximum velocity in each iteration is limited to half the decision variable search interval amplitudes.

The model obtained using the PSO algorithm is represented by:

$$G_{PSO}(s) = \frac{0.78}{1+152s} e^{-19.7s} \quad (7)$$

The results obtained from the open-loop step-response for both the first point, ($t_{35}=96.8s$, $y_{35}=39.9^\circ C$) and the second point, ($t_{85}=300.5s$, $y_{85}=70.3^\circ C$) representation, are observed in the top plot of Figure 5. In the bottom plot, the step response obtained with the two models is superimposed on the real data. As it can be observed, the mismatch between the PSO model and the real data is smaller than the $2pt$ model. This is also confirmed by the ISE values obtained: $ISE_{2pt}=555$ and $ISE_{PSO}=93$, for the $2pt$ and PSO models, respectively. Figure 6 presents the system output temperature and the simulation of one sample-ahead using the estimated FOPTD model for a step-input in closed loop. There

is a good agreement among the two responses, indicating that the model performs well.

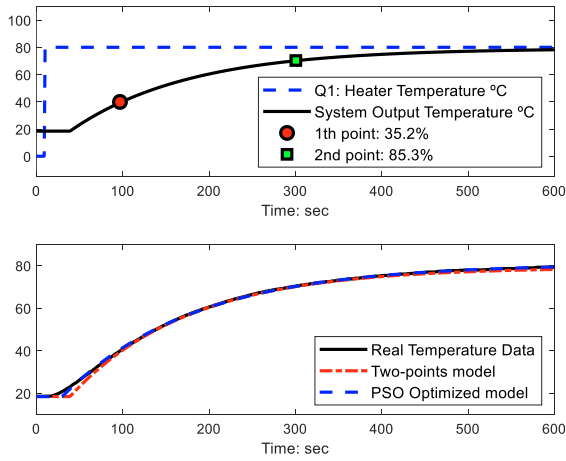


Fig. 5. Advanced TCLab open-loop step response and superimposed $2pt$ and PSO model responses over measured temperature data.

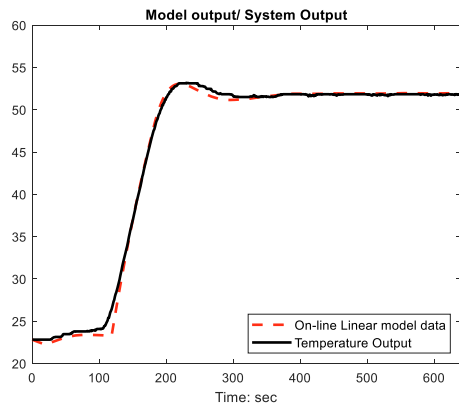


Fig 6. One sample-ahead FOPTD simulation over the measured temperature for a closed-loop TCLab test.

5. CONTROLLER DESIGN

The controller design is organized in the following control modes: On-off, Proportional (P), Proportional and Integrative (PI), Proportional and Derivative (PD) and Proportional Integrative and Derivative (PID). The controller’s designs are tested by students first by conducting simulations using Laplace based Simulink models. While the simulations can also be performed using digital-domain models, the simulation in the continuous-time domain enables students to validate the digital approximations used to implement the controllers in TCLab (for more info see Moura Oliveira and Hedengren, 2019). The proportional (P), Integrative (I) and Derivative (D) control actions are implemented in the digital domain, using the following approximations (absolute or position algorithm):

$$P_k = K_p e_k, I_k = K_i \sum_{i=1}^k \Delta T e_k, D_k = K_d \frac{y_k - y_{k-1}}{\Delta T} \quad (8)$$

with: e representing the error, k the sample index, K_p the proportional gain, $K_i = K_p/T_i$ the integrative gain, T_i the integral time constant, ΔT representing the sampling interval, $K_d = K_p T_d$ the derivative gain and T_d the derivative time constant. The PI, PD and PID controller are implemented using the parallel absolute (or

position) form, and the controller output for a PID controller is evaluated with (9). Note the derivative action applied only to the output signal to avoid the derivative kick effect when there is a change in set point.

$$u_k = P_k + I_k - D_k \quad (9)$$

For the sake of brevity, the following sections address only on-off control, PI control and PID control.

5.1 On-Off Control

The implementation and testing of a simple on-off control in the TCLab, is proposed to students. A test performed with the advanced TCLab controlled with an on-off controller is presented in Figure 7. In some cases (e.g. Arduino Uno basic TCLab) it is necessary to implement on-off control with a hysteresis band to avoid the unnecessary chattering between the on and off state (e.g. due to signal noise).

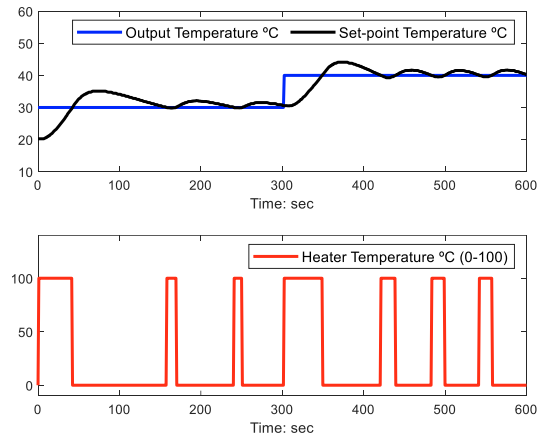


Fig 7. On-off control: test with the advanced TCLab.

5.2 PI Control

At this stage students have designed and tested P-only control and observe and evaluate the steady-state error (Moura Oliveira and Hedengren, 2019). Two main students learning issues regarding the introduction of the integral action over P-only control are: *i*) resetting the steady state error and improving the system response; *ii*) the integral windup problem and how to implement the anti-windup. This is made in the continuous time domain using a Simulink model and in the digital domain by halting the integration action when the heater is saturated. The difference of using or not the anti-windup scheme is observed in the test presented in Figure 8. Solid lines represent TCLab responses obtained using an anti-windup scheme (AW) and dashed lines represent the response without anti-windup. Without AW, it is clearly visible that the integral error accumulates when the heater is saturated. This makes the output temperature overshoot the target set point. Figure 9 presents the result of set-point tracking considering the initial set-point of 23°C and applying a step input with 35°C of amplitude at $t=100s$ and a load disturbance at $t=380s$ with -40°C of amplitude.

The ambient room temperature considered in the simulations is 18°C and in the TCLab tests ranges between 18°C to 23°C. Figure 9 top plot overlaps the simulated system output response in

the continuous time domain and the TCLab response. The simulation was performed using a Simulink Laplace based model (with fixed solver time interval of 1s). The sampling time used in all the digital TCLab tests was $\Delta T=1s$. Figure 9 (bottom plot) presents the heater control signals (simulated and from the TCLab test). The PI controller gains were evaluated using classical CC tuning rules, resulting in $K_p=8.9$ and $T_i=52.3s$. Note the good agreement between the simulation in the continuous time and discrete time-domains.

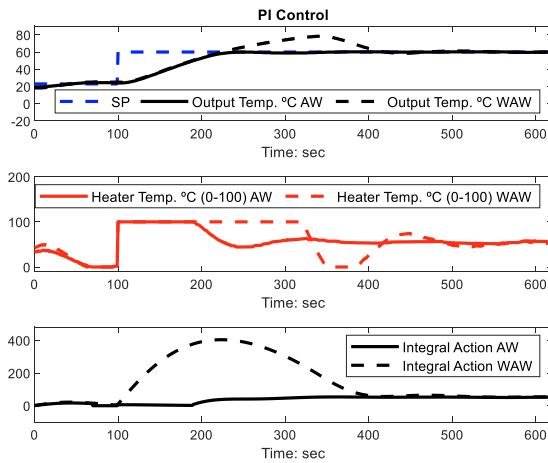


Fig. 8. PI Control: Comparison of TCLab responses: with anti-windup (AW) and without anti-windup (WAW).

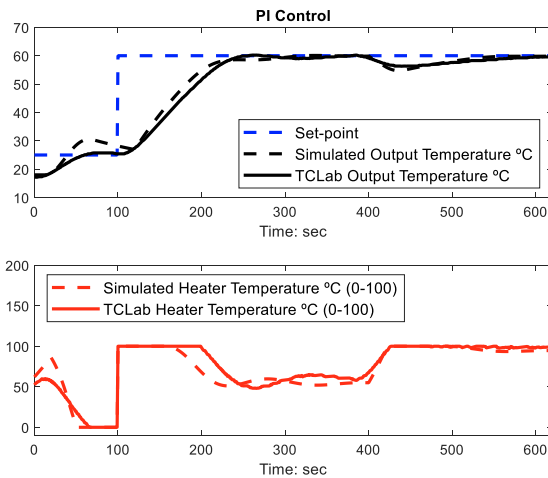


Fig. 9. PI Control: Simulated and TCLab responses.

5.3 PID Control

An important aspect students should apprehend when using PID control, is the importance of applying the derivative action to the controlled output signal instead of the error. The effect of derivative kick and its avoidance is illustrated in Figure 10 with a simulation of a setpoint change as shown in the top subplot and the difference in heater output shown in the bottom two subplots. The result of testing a PID controller tuned with the CC rules is presented in Figure 11. The parameters used in this test are the following: $K_p=13.5$ and $T_i=46s$ and $T_d=7s$. Compared with the PI response, the PID tracking performance is similar. However, regarding the load disturbance rejection, the PID performance is clearly better. It is important students observe the irregular behaviour of the control signal due to the derivative action.

Experiments incorporating a derivative action filter can also be implemented in TCLab to soften this behaviour.

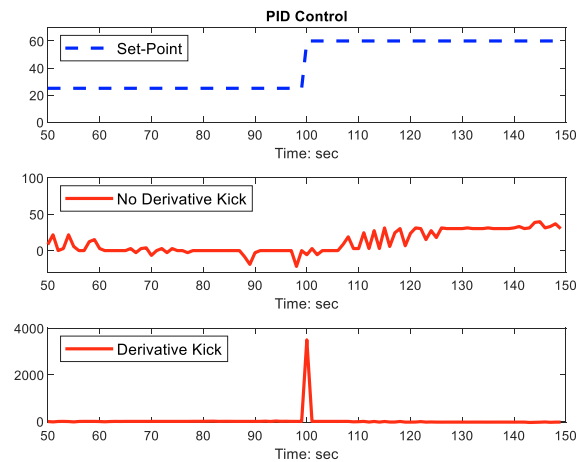


Fig. 10. PID derivative action: without and with derivative kick.

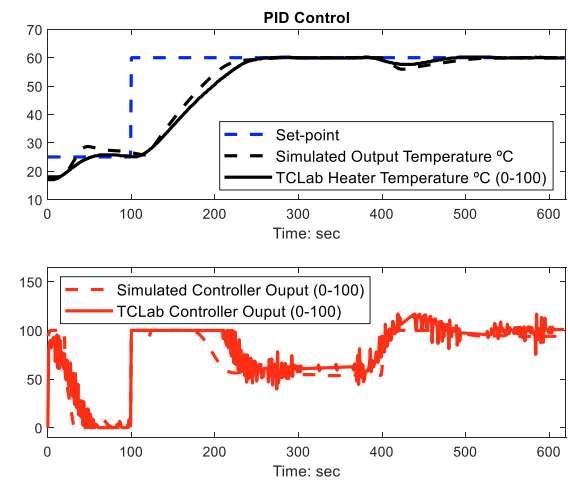


Fig 11. PID Control: Simulated and TCLab responses.

5.4 Experiment Execution Details and Students Perceptions

The start-up of this teaching experiment using TCLab temperature control occurred in the academic year of 2018-2019 in the Modelling and Control System course to the 3rd year of Biomedical engineering. The experience was improved and executed in the academic year of 2019-2020 to a similar undergraduate course to Bioengineering (see photo in Figure 12). The class has a small number of students: 17 and a TCLab is used by groups of two students (1 student choose to work alone). The experiment is designed to be executed by students in 3 consecutive practical classes (2h each). The experience was executed as follows: *i*) Class 1: first contact with TCLab; TCLab testing using the introductory Python programs described in section 3; Obtain an open-loop step response and export signal data to a text file be manipulated in MATLAB; Perform the FOPTD open loop identification using the $2pt$ and PSO method and compare the results (this last point can be completed out the class); *ii*) Class 2: Test the simple on-off controller; Design and test P and PI controllers (with and without anti-windup) *iii*) Class 3: Design and test PD and PID controllers. Students must deliver a written report with all the simulated and TCLab tests.



Fig 12. UTAD Bioengineering Students performing TCLab experiments.

Students engagement in practical classes with TCLab, both in last year and this year, was very positive. Some positive aspects outlined by students regarding the experiment are the following: *i)* simplicity of TCLab use; *ii)* the good agreement between the model real system as well as between controller simulations and TCLab test. Students did not like the time it took for each test (5-10 min).

6. CONCLUSION

An experiment to introduce digital industrial controller implementation to undergraduate students using the TCLab Arduino kit is reported. The control modes addressed with the experiment are: on-off, P, PI, PD and PID. The experiment involves performing the simulations using a MATLAB/Simulink environment and the TCLab experimental tests were programmed in Python. FOPTD model parameters are estimated from an open-loop step response using two methods: the two-point method and a PSO method. Controllers are tuning using Cohen and Coon rules. Simple experiments for the TCLab are proposed with potential to promote students understanding of PID control drawbacks, windup and derivative kick and how to solve each.

ACKNOWLEDGMENTS

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

REFERENCES

- Åström K. J. and Hägglund T., (2006), *Advanced PID Control*, ISA.
- Cohen G. H. and Coon G. A., (1953): "Theoretical consideration of retarded control", *Trans. ASME*, 75, pp. 827-834.
- Dorf R. C. and Bishop R. H. (2017), "Modern Control Systems", 13th Edition, Pearson Education.
- Fabregas E., Farias G., Dormido-Canto S., Dormido S. and Esquebre F., (2011), Developing a remote laboratory for engineering education, *Computers & Education* Vol. 57, pp. 1686–1697.
- Feedback Instruments, (2019), www.feedback-instruments.com/, retrieved in 10-11-2019.
- Feire H., Moura Oliveira P. B. and Pires E. J. (2017), From Single to Many-objective PID Controller Design using Particle Swarm Optimization, *Int. Journal of Control, Automation and Systems*, Vol. 15, Issue 2, pp. 918–932.
- Hedengren J. D. (2019), *Temperature Control Lab Kit*, <https://apmonitor.com/heat.htm>, retrieved in 10-11-2019.
- Hedengren J. D., Martin R. A., Kantor J.C., Reuel N. (2019) "Temperature Control Lab for Dynamics and Control", AICHE Annual Meeting, Orlando, FL, Nov 2019.
- Irigoyen E., Larzabal E. and Priego R., (2013) "Low-cost platforms used in Control Education: An educational case study", 10th IFAC Symposium Advances in Control Education, August 28-30, Sheffield, UK.
- Kantor, J. (2019), CHE 30338 Chemical Process Control with TCLab Jupyter Notebooks, URL: <https://jckantor.github.io/CBE30338/>, Retrieved: 10-11-2019
- Kennedy J. and Eberhart R. C., (1995): "Particle swarm optimization". In *Proc. of the IEEE Int. conf. on neural networks IV*, Piscataway, pp. 1942–1948.
- Moura Oliveira P. and Hedengren J. D., (2019), "An APMonitor Temperature Lab PID Control Experiment for Undergraduate Students", 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 790-797.
- Moura Oliveira P. B. (2018), "Design of Digital PID Controllers using Particle Swarm Optimization: A Video Based Teaching Experiment", 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control (PID), Ghent, Belgium, in *IFAC-PapersOnLine*, pp. 298-303.
- Park, J., Martin, R.A., Kelly, J.D., Hedengren, J.D., (2020) "Benchmark Temperature Microcontroller for Process Dynamics and Control", *Computers & Chemical Engineering*, Special Issue in Honor of Thomas F. Edgar, 135, doi: 10.1016/j.compchemeng.2020.106736.
- PID-18 (2019), Control Education Day, "What is an ideal undergraduate control curriculum", IFAC PID-18, Ghent, 10 May-2018, www.pid18.ugent.be/flierEDCO.pdf, retrieved in 10-11-2019.
- Prima E. C., Saeful K., Utarib S., Ramdanib R., Putri E. R. and Darmawati S. M., (2017) "Heat Transfer Lab Kit using Temperature Sensor based Arduino TM for Educational Purpose", *Procedia Engineering*, Vol. 170, pp. 536 – 540.
- Quanser, (2019), www.quanser.com/, retrieved in 10-11-2019.
- Reguera P., García D., Domínguez M., Prada M.A. and Alonso S. (2015), "A Low-cost Open Source Hardware in Control Education. Case Study: Arduino-Feedback MS-150", *IFAC-PapersOnLine*, Vol. 48-29, pp. 117–122.
- Rossiter J. A., Pope S. A., Jones B. L. and Hedengren J. D. (2019) "Evaluation and demonstration of take home laboratory kit", Invited Session: Demonstration and poster session, 12th IFAC Symposium on Advances in Control Education, July 7-9, 2019, Philadelphia, PA, USA.
- Seborg D. E., Edgar T. F., Mellichamp D. A. and Doyle III F.J., (2016), "Process Dynamics and Control", Wiley.
- Skogestad S. (2003), "Simple analytic rules for model reduction and PID controller tuning". *J. Process Control* 13:291–309.
- Silverstein D. L., Vigeant, M. A. and Staehle, M. (2016), "How We Teach Process Control: 2015 Survey Results". ASEE 123rd Annual Conference & Exposition, New Orleans, LA.
- Sundaresan K. R. and Krishnaswamy P. R. (1978), "Estimation of time delay, time constant parameters in time, frequency and Laplace domains", *Can. J. Chem Eng.*, 56, pp. 257.
- Visioli A. and Vilanova R., (Editors), (2012), "PID Control in the Third Millennium: Lessons Learned and New Approaches", *Advances in Industrial Control*, Springer Verlag