An APMonitor Temperature Lab PID Control Experiment for Undergraduate Students

Paulo Moura Oliveira INESC TEC, Department of Engineering Universidade de Trás-os-Montes e Alto Douro, UTAD 5001–801 Vila Real, Portugal <u>oliveira@utad.pt</u>

Abstract-Students born in a digital era require adjusted teaching and learning methodologies incorporating new technologies. A common difficulty found by students is how to test their controller designs in a real system. Thus, the development of affordable, portable and easy to use feedback control kits is highly desirable. The idea is that both lecturers and students can perform simple practical experiments anytime and anywhere. The APMonitor temperature control lab is an Arduino based control kit which fulfils these requirements. Proportional, integrative and derivative control is in operation in the vast majority of industrial process control loops. Thus, it is a mandatory topic in most undergraduate introductory feedback control courses. A teaching/learning PID control experiment for undergraduate Biomedical Engineering student's based on the temperature control lab is reported here. Results received from students are presented.

Keywords—PID Control, Control Engineering Education, Arduino.

I. INTRODUCTION

Introductory courses for control engineering require that students acquire knowledge and develop skills involving both theoretical and practical concepts. Most of these undergraduate courses require students to perform practical assignments using laboratory experiments (e.g. [1-3]). However, this type of laboratory equipment can be quite expensive. Thus, depending on the number of students and number of feedback control courses in a University, providing laboratory access to all courses is a challenging problem, particularly when resources are scarce. Moreover, providing students with extra class access to those practical laboratories is even more problematic. While the use of computer aided simulation software can attenuate the problem it can never replace the practical contact with the systems to be controlled. Remotely accessed laboratories, when available, can be an alternative. However, some of these experiments are rather complex and not very engaging for undergraduate students. So, easy to use and affordable control kits are highly desirable, both by lecturers and students. Even better is if the kit is small sized and can used anywhere (e.g. at home) and anytime. This is the case of the Temperature Control Lab (TCLab) based on an Arduino developed and proposed in

John D. Hedengren Department of Chemical Engineering, Brigham Young University, Provo, UT, USA john.hedengren@byu.edu

[4]. In this paper, a simple temperature control experience based on the TCLab is proposed, providing evidence of its simplicity of use.

Current students require new teaching and learning methodologies as well as tools. Many engineering students have just one control engineering course in their studies. This is the case for the first degree on Biomedical Engineering in UTAD in Portugal (three year program). Thus, currently an issue under debate can be expressed by the following question [5]: what is an ideal undergraduate control curriculum? Currently the vast majority of practical control applications are controlled with digital devices. However, many classical introductory feedback control courses do not include any digital control content. Thus, besides continuous time-domain control, it is also important to provide elementary digital control concepts to undergraduate students. The TCLab can also be used to consolidate digital control techniques.

It is well-known that the Proportional, Integrative and Derivative (PID) controller is widely used in most industrial control loops. Thus, the proper design and tuning of PID controllers is a relevant competency part of most feedback control curricula. Since the pioneering work developed by Ziegler and Nichols [6] many design and tuning rules [7] have been proposed for this type of controller. A significant number of tuning rules are evaluated using mathematical expressions based on a First Order Plus Time Delay (FOPTD) model. Indeed, for many commonly found processes (open-loop stable and non-oscillatory) it is possible to estimate FOPTD model parameters from an open-loop step response. Also for Biomedical control applications, many system dynamics can be approximated by FOPTD, such as the case of Mean Arterial Pressure control [8,9]. Examples of classical transient response identification techniques commonly taught to step undergraduate students are the Tangent and the Two-Point (2pt) [10] methods. The quality of this type of classical FOPTD models can be compared with models obtained with an optimization method.

The experiment reported here has the following learning objectives:

- How to identify a FOPTD model using a step response classical method?
- How to identify a FOPTD model using a step response optimization method?
- What are the main characteristics of the PID actions, considering the following combinations: P, PI, PD and PID?
- How to design PID controllers using tuning rules?
- How to digitally implement a PID controller?

The remaining of the paper is organized as follows: section II presents a brief description of the TCLab. Section III is dedicated to identification of a FOPTD. Section IV presents P, PI, PD and PID tuning and testing results. Section V presents some feedback provided from students regarding the proposed experiment. Finally, section IV concludes the paper.

II. APMONITOR TEMPERATURE CONTROL LABORATORY OVERVIEW

The TCLab (see Fig. 1 and Fig. 2) is based on an Arduino, with two heaters and two temperature sensors. Thus it was designed to perform experiments based on uni-variate and multivariate temperature control. The TCLab is connected to the computer through a USB serial port and requires power to the heaters using an electrical plug.



Fig. 1. APMonitor Temperature Control Lab (TCLab) kit schematic (extracted from [4]).

The TCLab allows a user to perform control temperature experiments using several Process Control techniques with computer coding either in Python or Matlab. Moreover, background information involves Scripts, Videos and tutorials that are freely available in [4]. TCLab is a portable kit which can easily be used anywhere with computer and an electric power connection.



Fig. 2. APMonitor Temperature Control Lab (TCLab) kit components (extracted from [4]).

III. FOPTD MODEL IDENTIFICATION

Students can easily test several open-loop step response identification models by simulation. The objective here is that students perform a FOPTD model identification using real open-loop step response data. The TCLab open-loop response with just one operating heater is presented in Fig. 3. The step input was applied at t=10s.



Fig. 3. Open-loop step response obtained from TCL basic kit.

As stated in the previous section, students have to apply both a classic method and an optimization method. The classic method reported is the Two-Point method [10]. While the optimization model can be determined using any optimization technique [4], in this experiment a canonical Particle Swarm Optimization (PSO) [11] was deployed. The PSO is a nature inspired algorithm mimicking swarm behavior. It has been incorporated in several control education experiments by the first author [12,13] with good results. Indeed, the blending of Artificial Intelligence in designing control systems is well accepted by students. For more information regarding using the PSO within PID controller design refer to [13].

A. Two-Point Method

Here students have to code a script (in Python or Matlab) to determine the first point (35.2%) and second point (85.3%) of the transient system output response, which in this case is the measured temperature. Representing both point coordinates as (t_{35} , y_{35}) and (t_{85} , y_{85}), in this case these correspond approximately to (84.32s, 30.05°C) and (272.4s, 47.41°C), respectively. The two points are represented in Fig. 4.



Fig. 4. Two-point representation in the open-loop step response.

The temperature steady-state value, y_{ss} , and initial value, y_{0} , must also be determined, which in this case are: $y_{ss}=52.51^{\circ}$ C and $y_{0}=17.8^{\circ}$ C. Thus applying the following formula:

$$K = \frac{y_{ss} \cdot y_0}{u_{ss} \cdot u_0} = 0.35 \tag{1}$$

$$T = 0.67(t_{85} - t_{35}) - t_0 = 116s$$
⁽²⁾

$$L = (1.3t_{85} - 0.29t_{35}) - t_0 = 20.8s$$
(3)

results in the following 2pt FOPTD model:

$$G_{2pt}(s) = \frac{0.35}{1+116s} e^{-20.8s}$$
(4)

B. Optimization Model

A simple Matlab PSO script was provided to students to perform the FOPTD model optimization. The following PSO conditions were used in the tests: 40 particles, evolved during 50 iterations, using an inertia weight linearly decayed in the interval [0.7,0.4]. The minimization criterion was the sum of the square of the error between the model and the real data test corresponding to Fig. 3 step response. The PSO search was conducted assuming a parameter range for *K*, *T* and *L* of [0.13], $[20s\ 160s]$ and $[4s\ 45s]$, respectively. The resulting optimized model is represented by (5). The simulated step responses of both identified models, 2-point and optimization, are overlapped on the real process temperature data in Fig. 5. As it can easily be observed, the optimized model match to the real temperature data is superior to one achieved with the 2pt model. Students can also confirm this performance difference by evaluating the *ISE* value considering the error mismatch resulting in approximately: $ISE_{2pt}=171$ and $ISE_{opt}=50$, for the 2pt and optimization (*opt*) model.

$$G_{opt}(s) = \frac{0.35}{l+1.31s} e^{-16.3s}$$
(5)



Fig. 5. Overlapped step responses for the 2pt and optimized model over real TCLab temperature data (basic kit).

IV. PID CONTROL

Based on the best performing FOPTD model estimated in the previous section, students design several controllers with different control modes, namely: P, PI, PD, and PID. As stated previously, there are hundreds of tuning rules for PID controllers. The experiment aim is not to use the best performing rules. Students were provided with some classical PID tuning methods, such as: Cohen-Coon (CC) [14], IAE [15] and AMIGO [16]. The CC method results are reported here. After tuning and simulating the respective controller in the continuous time domain the objective is to test and validate the controller in the TCLab. This involves using a digital controller implementation. In this case the controller was implemented in Python following the APMonitor script available in [4]. Note that all the results presented in this section were obtained using a basic TCLab kit.

A. Proportional Control

The results obtained with the CC setting for proportional control resulted in a proportional gain of Kp=23.9. The simulated test response is presented in Fig. 6 and the corresponding TCLab in Fig. 7. All simulations were performed with Simulink considering a fixed sample time of 1s for the solver, which matches the sampling period used with the digital control. The following actuator limits were considered $(0 \le u(t) \le 100)$. The most important thing students should comprehend with these tests is the practical impossibility to eliminate the steady state error with proportional control. The simulated step results are in accordance with the real TCLab response.



Fig. 6. P-only control with CC settings: simulation test.



Fig. 7. P-only control with the CC settings: TCLab test.

B. Proportional and Integrative Control

The results obtained with the CC setting for PI control resulted in a proportional gain Kp=20.9 and integrative time constant of Ti=41.1s. Due to the saturation limits it is important to incorporate an anti-windup (AW) scheme [17] both in the continuous and in the digital time domain. The anti-windup scheme used in the continuous time domain is represented in Fig 8, considering a PID controller. With PI control the derivative action applied to the system output was excluded.



Fig. 8. Anti-windup scheme used in simulations

The digital PI controller is governed by the difference equation represented by (6) with, *e* representing the error, $K_i = K_p/T_i$ the integrative gain, ΔT representing the sampling interval, and *P* and *I* the proportional and integral actions, respectively.

$$u_k = K_p e_k + K_i \sum_{i=1}^k \Delta T e_k = P_k + I_k \tag{6}$$

The simulated test response is presented in Fig. 9 and the corresponding TCLab in Fig. 10. As it can be observed there is a good match between the simulated and real responses. Students should observe the steady-state error elimination, as a major advantage of *PI* over *P*-only control.



Fig. 9. PI control with CC settings: simulation test.



Fig. 10. PI control with the CC settings: TCLab test.

It important that students realize the integrative anti-windup scheme role. A simple test can be performed with the TCLab by not resetting the integrative action when the actuator is saturated. A step response obtained with the same CC PI settings is presented in Fig. 11. It shows clearly the role of the anti-windup and the performance improvement achieved.



Fig. 11. Comparison of PI with AW and without AW: TCLab test.

C. Proportional and Derivative Control

The results are obtained with the CC setting for PD control resulting in a proportional gain Kp= 29.1 and derivative time constant of Td= 4.2s. The digital PD controller is governed by the difference equation (7) with, *e* representing the error, *y* the system output, $K_d = K_p T_d$ the derivative gain, ΔT representing the sample interval, and *P* and *D* the proportional and integral actions, respectively. It is relevant that students note that the derivative action is applied to the system output, rather than to the feedback error. This is to avoid the derivative-kicking effect.

$$u_k = K_p e_k - K_d \frac{y_k - y_{k-1}}{\Delta T} = P_k - D_k \tag{7}$$

The simulated test response is presented in Fig. 12 and the corresponding TCLab in Fig. 13. There are strong similarities between the simulated and real signals trend. There is also a significant difference in the real control signal compared with the simulated due to the derivative action. This control signal variation can be attenuated by using a filter in the derivative action. Note the non-null steady state error.



Fig. 12. PD control with CC settings: simulation test.



Fig. 13. PD control with the CC settings: TCLab test.

D. Proportional, Integrative and Derivative Control

The results obtained with the CC setting for PID control resulted in a proportional gain Kp=31.3, integrative time constant Ti=38.2s and derivative time constant Td=5.8s. The digital PID controller is governed by the difference equation (8)

$$u_{k} = K_{p} e_{k} + K_{i} \sum_{i=1}^{k} \Delta T e_{k} - K_{d} \frac{y_{k} - y_{k-1}}{\Delta T} = P_{k} + I_{k} - D_{k}$$
(8)

The simulated test response is presented in Fig. 14 and the corresponding TCLab in Fig. 15.



Fig. 14. PID control with CC settings: simulation test.

Fig. 16 presents the overlapping step responses obtained with PI and PID controllers. Comparing the PID and PI response with the CC settings, no improvement it noticeable with the inclusion of the derivative action from the PID to the PI controller with the CC settings. However some tuning rules, such as [18], do not recommend derivative action for FOPTD systems.



Fig. 15. PID control with the CC settings: TCLab test (basic kit).



Fig. 16. Overlaped TCLab responses for PI and PID control (basic kit).

V. EXPERIMENT FEEDBACK FROM STUDENTS

The reported experiment was tested in a UTAD university Modelling and Control Systems course held in the first semester of the 2018-2019 academic year. The respective work was conducted in three consecutive laboratory classes (2h each), with students delivering a written report afterwards. There were 16 students conducting the experiment and organized in groups. Students performed their simulation mostly using Matlab scripts and Simulink models. However, the digital implementation to perform the TCLab tests used Python scripts provided by [4]. Tests data was stored in texts files and processed in Matlab. In the case of the FOPTD identification, data corresponding to the step response presented in Fig. 3 was also provided to students. The data file can be downloaded from [19] for experiment replication. Not excluding the step-responses using the TCLab performed by students, this allowed all students to use the same step-response data for model performance comparison.

Two TCLab kits used by students in the practical classes were an advanced control lab [4]. This kit is different from the basic one used to perform the test results presented in section IV. The advanced TCLab uses a different power supply and Arduino Leonardo to improve the connection experience for plug and play access [4]. Thus, the results obtained by these groups of students were different for the groups using the basic one. The identified FOPTD model is different and consequently in the corresponding designed controllers. The results obtained for the advanced TCLab regarding the FOPTD model identification are illustrated in Fig. 17 by the overlapped step responses between the real data, 2pt and PSO optimized models. In this case the 2pt model parameters are: K_{2pt} =0.75, T_{2pt} =123.1s, and L_{2pt} =23.4s while the PSO optimized model parameters are: K_{opt} =0.75, T_{opt} =145.9s, and L_{opt} =19.5s.



Fig. 17. Overlapped step responses for the two-point and optimized model over real TCLab temperature data (advanced kit).



Fig. 18. PID control with the CC settings: TCLab test (advanced kit)

The CC settings for this system (advanced TCLab) are the following: Kp=13.6, Ti=45.5s and Td=6.9s and the corresponding set-point tracking response is presented in Fig. 18. The CC tuning gains for this case result in a different performance when compared with the corresponding results obtained with the basic TCLab (see Fig. 15). The system response is faster, however the tracking for the first step is worst and there is an overshoot for the second step. However, students

clearly noticed that the control signal variation obtained with the advanced TCLab is clearly smaller than the one obtained with the basic TCLab. This is also noticeable by comparing Fig. 18 with Fig 15 signals.

The oral feedback received from students regarding the experiment in class was quite positive. Student's appreciated the following aspects:

- To be able to compare the simulation with the real TCLab results, obtaining a good correspondence between them. These provided evidence that the linear FOPTD model was well identified and that there are a good agreement among the continuous and digital domain PID control tests. Also with the used Pyhton software [4] students could visualize the on-line tracking as well as the good matching between the real temperature and the model predicted temperature.
- The easiness of use of the TCLab (requires installing the appropriate Arduino software). The simple procedure of plugging the USB cable to the computer, powering the heater, and using the TCLab was easy to follow.

VI. CONCLUSION AND FURTHER WORK

A new teaching and learning experience for an undergraduate introductory feedback control courses was proposed. It addresses a practical control problem by using the APMonitor Temperature Control Laboratory (TCLab). The control engineering topics addressed are: i) FOPTD model identification based on an open-loop step response. ii) P, PI, PD and PID design using classical tuning rules (Cohen-Coon). iii) Two major PID controller's limitations: derivative-kicking and integral anti-windup and how to avoid it. iii) The simple digital PID control implementation, based on the position (or absolute) algorithm. Both the identification and control simulations were carried out using Matlab scripts and Simulink models. The FOPTD models were identified based on a TCLab open-loop step-response using the two-point technique and an optimization tool (Particle Swarm Optimization). The best performing model was used to design P, PI, PD and PID controllers, which were then tested in the TCLab.

The experiment was carried out by undergraduate UTAD University Biomedical Engineering students in the first 2018-2019 semester. The student's feedback regarding the proposed experience and about using the TCLab was quite positive. The positive results achieved with TCLab motivate further validation the proposed experiment. More student's results are necessary by conducting a pedagogical questionnaire inquiry. The current experiment can be extended by addressing the following topics: i) The design of PID controller using optimization tools; ii) disturbance rejection response analysis; iii) Testing other anti-windup schemes; iv) Implement on-off control; vi) implement other PID control schemes, etc.

ACKNOWLEDGMENT

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project «POCI-01-0145-FEDER-006961», and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

- Feedback Instruments, <u>http://www.feedback-instruments.com/</u>, retrieved in 7-4-2019.
- [2] Quanser, https://www.quanser.com/, retrieved in 7-4-2019.
- [3] Gunt Technology, <u>http://www.gunt-technology.co.uk/products/</u>, retrieved in 7-4-2019.
- [4] J. Hedengren, Temperature Control Lab Kit, http://apmonitor.com/heat.htm, retrieved in 7-4-2019.
- [5] Control Education Day, "What is an ideal undergraduate control corriculum", IFAC PID-18, Ghent, 10 May-2018, <u>http://www.pid18.ugent.be/flierEDCO.pdf</u>, retrieved in 7-4-2019.
- [6] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers", Transaction of the ASME, pp. 759-768, 1942.
- [7] A. O'Dwyer., Handbook of PI and PID Controller Tuning Rules (2nd Edition), Imperial College Press, ISBN 1-86094-622-4, 2006.
- [8] J. F. Martin, A.M. Schneider and N. T. Smith, "Multiple-model adaptive control of blood pressure using sodium nitroprusside", IEEE Trans Biomed Eng., 34, No.8, Vol. BME-34, pp. 603-611, 1987.
- [9] R. W. Jones and M. T, Tham, "An Undergraduate CACSD Project: the Control of Mean Arterial Blood Pressure during Surgery", Int. J. Eng. Ed. Vol. 21, No. 6, pp. 1043-1049, 2005.
- [10] K. R.Sundaresan and P. R. Krishnaswamy (1978), "Estimation of time delay, time constant parameters in time, frequency and Laplace domains", Can. J. Chem Eng., 56, pp. 257.
- [11] J. Kennedy and R. C. Eberhart: "Particle swarm optimization". In Proc. of the IEEE Int. conf. on neural networks IV, Piscataway, pp. 1942–1948, 1995.
- [12] P. B. Moura Oliveira and J. C. Boaventura, "Blending Artificial Intelligence into PID Controller Design: A Biomedical Engineering Experience", 11th IFAC Symposium on Advances in Control Education, Bratislava, Slovakia, IFAC-PapersOnLine 49-6, pp. 366–371, 2016.
- [13] P. B. Moura Oliveira, "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, May 09-11, 2018.
- [14] G. H. Cohen and G. A. Coon: "Theoretical consideration of retarded control", Trans. ASME, 75, pp. 827-834, (1953).
- [15] P. W. Murril, P. D. Schnelle Jr., B. G. Lipták, J. Gerry, M. Ruel and F.G. Shinkey, "Tuning PID Controllers", Chapter 2.25, Ed. Lipták B. G. Process Control: Instrument Engineers Handbook, Butterworth-Heineman, 1995.
- [16] K. J. Åström and T. Hägglund, "Revisiting the Ziegler–Nichols step response method for PID control", Journal of Process Control, Vol. 14, pp. 635–650, 2004.
- [17] K. J. Åström and T. Hägglund, Advanced PID Control, ISA, 2006.
- [18] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning". J. Process Control 13:291–309, 2003.
- [19] P. B. Moura Oliveira, "data_step_3.txt". Open-loop step response <u>https://www.dropbox.com/s/7i7zpqhmi4o570p/data_step_3.txt?dl=0</u>, retrieved in 12-4-2019.