TEMPERATURE AND MASS FLOW CONTROL OF AN EDUCATIONAL AIR BLOWER

JOSÉ CARLOS R. DE OLIVEIRA
DEPART. DE ENGENHARIA ELETRÔNICA, UNIVERSIDADE FEDERAL DE MINAS GERAIS
AV.ANTÔNIO CARLOS, 6627 – 31270-901 - CAMPUS PAMPULHA, BELO HORIZONTE, MG
JCARLOSR@UFMG.BR

YASSIN LAJNEF
ESIEE – ÉCOLE SUP. D'INGENIEURS EN ÉLECTRONIQUE ET ÉLECTROTECHNIQUE
PARIS, FRANCE
YASSIN.LAJNEF@EDU.ESIEE.FR

Abstract— This paper describes an educational air blower, a reduced-scale process for teaching instrumentation and control as well as fluid dynamics. Constructed from a commercial hair-dryer, its automation is made by an Arduino microcontroller, with the aid of air temperature and speed sensors, mass flow being estimated by real-time software. Arduino performs the real-time tasks, while a human-machine interface is implemented on a Matlab console, in a personal microcomputer, this whole system permitting the students to control the blower in open-loop or closed-loop. Black-box identification and model validation are conducted for both temperature and mass flow. Experimental results complete the work.

Keywords— Laboratory educational kit, air blower, temperature control, mass flow control.

1 Introduction

A blower is a mechanical device used to pump air or other gases, constituted by a fixed element and a rotor to increase the fluid pressure, furnishing kinetic energy and forcing it to move against a mechanical resistance of ducts, dampers, etc. If it has a heating part, thermal power is transferred to the fluid.

Air blowers have application in building's ventilation, in central heating systems and in ambient cooling. In industry, commerce and warehouses, they are normally constituted by an element placed in the direction of flow to heat the air. They can be applied as dryers, with axial or centrifugal ventilators.

In combustion processes the air is the source of oxygen and, as it is fundamental to know the right composition of all inputs, chemical stoichiometry demands data for the combustion calculations. When furnishing air to furnace burners, its control is crucial: mass flow influences the right proportion between oxygen and fuel, and a higher air temperature increases the efficiency of the process. The air density varies according to the atmospheric pressure (altitude), as well as to temperature, so it is necessary to correct these variables in density estimation.

The developed prototype of the Air Blower is shown in figure 1.

Figure 1. The air blower prototype

The objective of this work is to provide an educational reduced-scale process for teaching instrumentation and control, as well as fluid dynamics, proportioning a real contact with equipment similar to the industrial ones.

The process was implemented from a commercial 1,200 W hair-dryer, but its two conventional speed and heating commands have been suppressed and substituted by two independent AC converters. The pump is driven by a DC series motor ("universal motor"), which accepts either a DC or an AC source. In this way, the air speed and the heating power may be continuously adjusted, from zero to their maximum, through the variation of the correspondent thyristor firing angles. A tube was coupled to the hair-dryer, increasing its length, and an aluminum tower was added to strengthen the whole structure.

Each of the AC converters has two back-to-back SCRs (silicon controlled rectifiers), which are successively turned on, one at the AC line positive half-cycle, and other at the negative one, with a proper delay angle (Cortizo, 2006). Thus, they apply RMS controlled voltages to their charges, e.g., the DC series motor (for blowing) and the resistive heater.

The blower automatic control is implemented by an 8-bit Arduino Mega 2560 microcontroller, programmed in C language into an IDE-Integrated Development Environment (Arduino, 2015) and connected to a PC microcomputer through a serial USB cable. The process is operated at the PC, by a console developed in Matlab (Mathworks, 2015). Temperature / air speed sensors were added.

2 Transducers and analog-to-digital conversion

The air temperature transducer was implemented from a P-N junction sensor, by a signal metal-package transistor: for 0^o Celsius, its base-emitter voltage is approximately 0.7V; however, as in this case we want the output transducer to be 0.0 V, we have introduced an off-set by means of a potentiometer, to make this "zero" adjustment. Then we amplify this voltage to have 5.0 V for 100˚C. The experimental static characteristic curve is showed in figure 2, and figure 3 presents the linearity error from the difference between measured and theoretical data.

As we can see, the temperature transducer presented a good linearity, with errors smaller than 1.0% of the span. The transducer can be modeled by:

$$\boldsymbol{Vt = k_t \times Tc} \tag{1}$$

with:

$$\boldsymbol{k_t = 4.98 \times 10^{-2} V/℃}$$

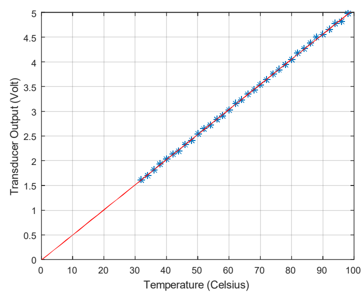

Figure 2. Temperature static characteristic

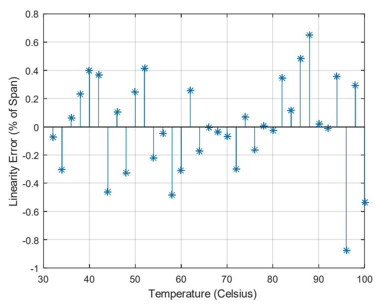

Figure 3. Temperature linearity error

For mass flow estimation, one must measure the air medium speed inside the blower tube. Then, mass flow is calculated in real-time by software, taking into consideration the air density, corrected by the temperature and by the local atmospheric pressure. For the speed, a mini-cooler (used in microcomputer motherboards) was inserted in the medium portion of the tube, not functioning as a motor, but in turbine mode, its angular speed being proportional to the air medium speed (Fox, 2001). Fed by a lower voltage, necessary only to excite its sensor part, the cooler output is a square-wave signal, whose frequency varies according to the rotor angular speed. Then we use a frequency-to-voltage converter, obtaining 0.0 V for zero air speed and 5.0 V for maximum speed.

The frequency experimental static characteristic is shown in figure 4. The linearity error was less than 1.0% of the span.

The frequency sensor can be modeled by:

$$\boldsymbol{Vf = k_f . f} \tag{2}$$

with:

$$\boldsymbol{k_f = 1.97 \times 10^{-2} V/Hz}$$

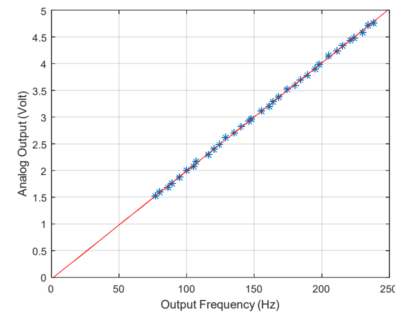

Figure 4. Frequency static characteristic

The calibration of the air speed transducer was done with a laboratory anemometer, thus giving the final characteristic presented in figure 5.

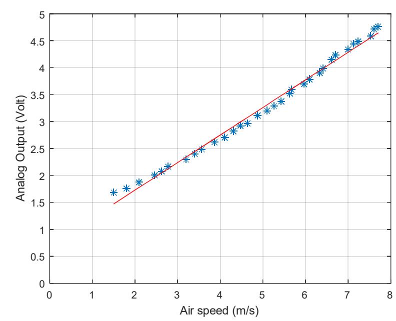

Figure 5. Air speed static characteristic

The air speed estimator was modeled by:

$$\boldsymbol{Vv = k_v . \overline{v}} \tag{3}$$

with:

$$\boldsymbol{k_v = 0.6027\ V.s/m}$$

The linearity error was less than 3.0% of the span. For small speeds, the calibration with the anemometer was not as good as for higher ones. Analyzing figure 5, we *presume* that this greater linearity error and the off-set in the *estimated* speed were due mainly to the anemometer used in calibration. So, the blower will be operated in medium / high speeds.

The signals from the temperature and air speed transducers are furnished to the Arduino microcontroller through two of its 10-bit analog-to-digital converters, giving a resolution of 1/1024, which we consider adequate for the blower application.

3 Mass flow estimation

The **volumetric flow** is:

$$\boldsymbol{q_v = A.\overline{v}} \tag{4}$$

where $\boldsymbol{\overline{v}}$ *is the medium value of the air speed* and $\boldsymbol{A}$ *is the inner area of the blower tube*. As we have a radius of 2.6 cm:

$$\mathbf{A = \pi \times r^2 = 2.19 \times 10^{-3}\ m^2}$$

then:

$$q_v = 2.19 \text{ x } 10^{-3}\bar{v} \tag{5}$$

The **mass flow** is defined according to the expression:

$$q_M = \rho . q_v \tag{6}$$

where q_v *corresponds to the volumetric flow and* $\boldsymbol{\rho}$ *corresponds to the air density,* calculated by (Sonntag, 2003):

$$\rho = \frac{p}{R.T_K} \tag{7}$$

Finally, for mass flow, from (4), (6) and (7):

$$q_M = \frac{p.A.\bar{v}}{R.T_K} \tag{8}$$

p corresponding to the atmospheric pressure which is, in Belo Horizonte, Brazil, 854 m above the sea (INMET, 2016):

$$p = 92{,}030 \; Pa$$

The mass flow is proportional to air speed and pressure and inversely-proportional to temperature. T_K *is the temperature in Kelvin and* T_C in Celsius degrees:

$$T_K = T_C + 273.15 \tag{9}$$

R *is the constant of gases applied to the air*:

$$R = 287.0530 \text{ J} \cdot \text{kg}^{-1} \text{ . K}^{-1}$$

So, we have:

$$\rho = \frac{92{,}030}{287.0530 \times T_K} \tag{10}$$

Finally:

$$q_M = 2.19 \text{ x } 10^{-3}.\bar{v}.\frac{92{,}030}{287.053 \times T_K} \tag{11}$$

$$q_M = \frac{0.7021.\bar{v}}{T_K} \tag{12}$$

4 Digital-to-analog conversion

To drive the blowing and heating powers with Arduino Mega 2560, as it doesn't have any DAC (Digital-to-Analog) option, we have used two of its 8-bit PWM (Pulse Width Modulation) outputs, passing these signals through two Sallen and Key low pass active filters (Sedra,1998). In this manner, we have recovered their analog average values, in a range of 0-10V, to use as inputs for the two thyristor-based AC drives. Figure 6 presents the static characteristic of one of these "DAC" converters, their linearity error being less than 1.0%.

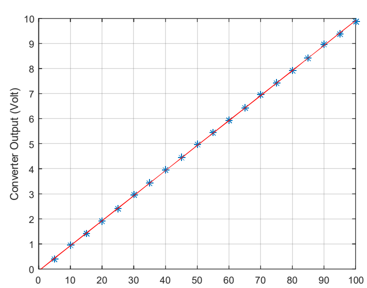

Figure 6. DAC static characteristic

5 Blower control architecture

Theoretically, two methods can be used to control the blower. The simplest is using Matlab library, Arduino being applied for serial communication with Matlab/PC and for I/O interface with the blower. The second alternative is making Arduino do the real-time tasks, Matlab/PC performing the human-machine interface (HMI) to enter references and parameters for the blower operation, as well as for visualization of the controlled variables. A bottleneck in the first option makes us choose the second one: the serial communication between the two mentioned parts is not deterministic, so a variable time-lag will appear and deteriorate real-time operation. In this manner, a console was built on the Matlab environment (Lajnef, 2015), constituting a GUI (Graphical User Interface), as shown in figure 7.

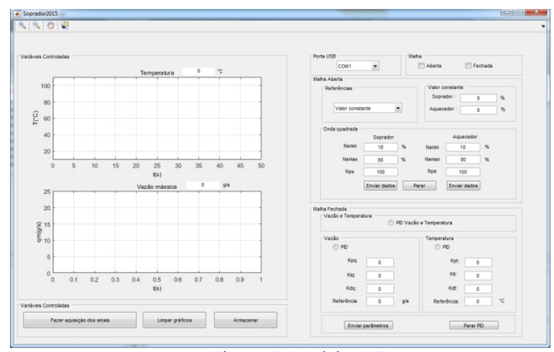

Figure 7. Matlab GUI

The console permits the students to control the blower in open and in closed-loop. It acquires data and displays them, drawing (in real-time) the curves of air temperature and mass flow versus time, also showing them numerically. It is also possible to save data in the microcomputer hard disk.

At open-loop, one can independently establish 0 to 100% blowing and heating voltage commands, or apply a step to one of these inputs, maintaining the other constant, for black-box process identification.

It is possible to operate the blower closing only one loop, temperature or mass flow, and also both, the user entering each three PID gains and set-point.

The Arduino C language programming and compiling were made at the PC, and the machine-code transferred through their serial connection.

The communication between Matlab and Arduino, as well as the main tasks implemented by each one, are shown in figure 8. The communication is bi-directional. Matlab (on the PC) is charged of sending the reference values for open and closed-loop control and the related parameters (established by the user) and receiving the two controlled variables (temperature and mass flow) to be displayed or saved. Arduino implements the PID controllers for closed-loop and the PWM actions for the process, sending back the controlled variables to the PC (Lajnef, 2015).

Figures 9 and 10 present the schematics for the temperature and mass flow closed-loops. Besides

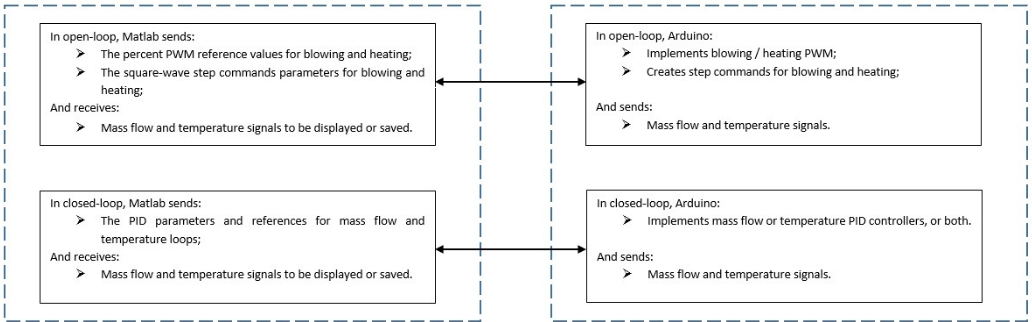

Figure 8. Communication Matlab – Arduino

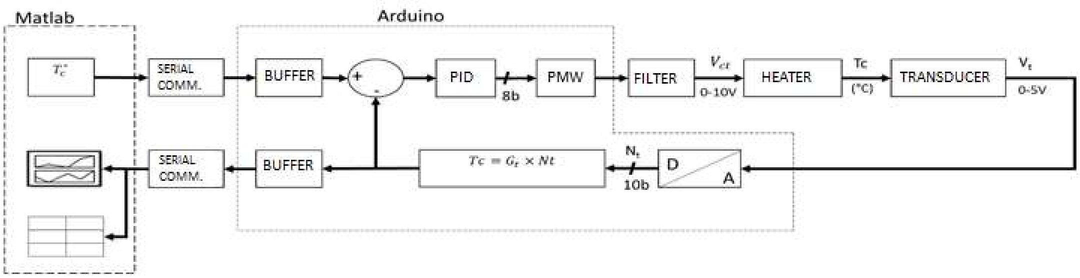

Figure 9. Schematic for temperature closed-loop

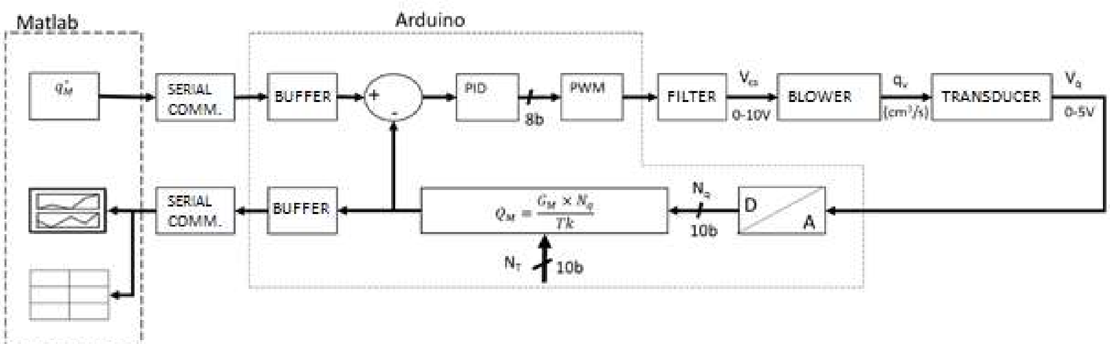

Figure 10. Schematic for mass flow closed-loop

the PID parameters, the set-point inputs are T_C^* for temperature and q_M^* for mass flow.

After receiving data, Arduino compares each set-point to the respective measured / estimated controlled variable, and implements a PID algorithm. Then, it charges the PID output on an 8-bit PWM (Pulse Width Modulation) internal module, whose output is transformed into a signal compatible to the thyristor converter. This is done independently for the blowing and the heating parts of the process.

The analog acquired and processed temperature and air speed are fed back from the transducers to Arduino, in a 0-5V format, through its analog-to-digital converters, and finally processed numerically. For temperature, it is only necessary to apply a proper conversion gain but, for mass flow, equation (12) must be used, with the correspondent temperature and air speed conversion factors.

Finally, Arduino transmits air temperature and mass flow serially back to the PC console, to be displayed numerically and graphically, and also to be saved in a file.

6 Process safeguarding

Two kinds of protection were implemented on the air blower prototype: a hardware one, to act immediately on the SCRs low-level heating command, and a software high-level version. Both permit the application of power on the heater circuit only if at least 50% of air pumping action takes place. This is necessary because the blower (the commercial hair-dryer) was dimensioned to dissipate heat in a forced manner.

7 Experimental results

7.1 Black-box identification and model validation

Figures 11 and 12 show the responses, respectively due to 10% up and 10% down step commands around a 50% heating operating point (blowing at 50%). A first order plus time-lag linear model was identified from the rising experimental curve (fig. 11), which was used to simulate the falling step response and compare it to the experimental one (fig.12) for a validation purpose (Aguirre, 2007). Off-sets were introduced on the curves to make signals start/finish from/at zero.

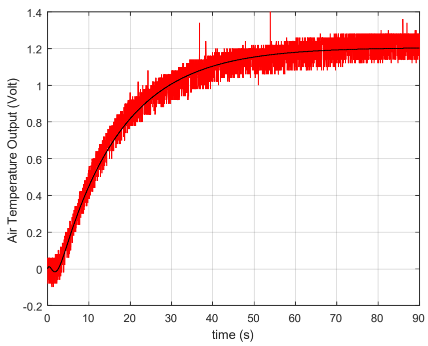

Figure 11. Air heating rising step response

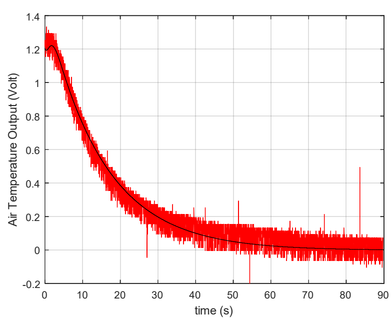

Figure 12. Air heating falling step response

Figures 13 and 14 present the responses, respectively due to 10% up and 10% down step commands around a 60% blow operating point, with 0% heating. Again, a first order plus time-lag linear model was identified from the rising experimental curve (fig. 13), which was used to simulate the falling step response (fig.14). For the same reason, off-sets were introduced.

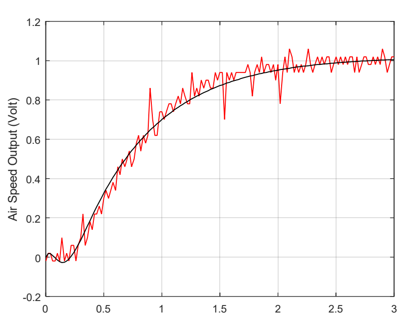

Figure 13. Air speed rising step response

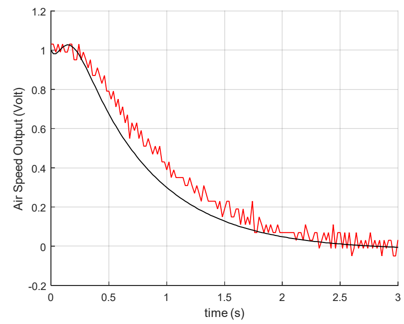

Figure 14. Air speed falling step response

Analyzing the four precedent figures, we see that the model for the heating action, obtained from the rising response, fits very well the falling curve. On the other hand, the model for blowing is not so good for the experimental falling curve, which is slower than the rising one, showing a non-linear dynamics at the considered operating point. We can also see that the temperature dynamics is much slower than the air speed one. These models could be used by the students for tuning the temperature and mass flow PIDs.

7.2 Closed-loop experimental results.

As discussed before, the blower can be operated with only one closed-loop, temperature or mass flow, or with both. The Matlab console permits to enter the PID gains and the set-point for each loop. Figures 15 and 16 show the two controlled variables and their set-points, in real-time operation. The first presents the temperature response to a step (up), practically showing no effect on mass flow (down). Figure 16 shows the response to a step in mass flow (up) and the correspondent profile of temperature (down, divided by 10).

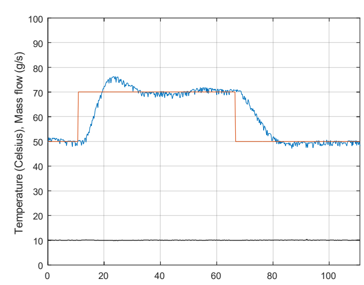

Figure 15. Response to a step in temperature

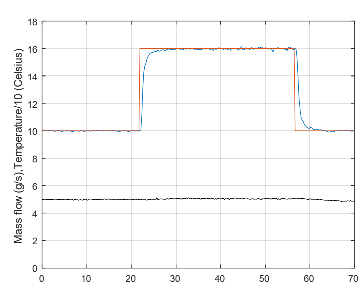

Figure 16. Response to a step in mass flow

ISSN 2525-8311

In each test, the perturbation given by a strong transition in one controlled variable is quickly corrected by the other loop, although there is a coupling between them. The temperature loop maintains its slower time response and is more critical to control, as it oscillates a little around its medium, with a ripple due to the transducer noise.

In accordance to Aström and Hägglund (2001), the PID controller is not outdated, and it can be combined with strategies such as Smith predictors, gain scheduling, filters, fuzzy control and others, to face non-linear and more complex processes. In this work, we have obtained open-loop deterministic models for temperature and mass flow, but no model-based method was used for tuning the PID controllers, as the purpose of this paper was only to present the potential of the educational blower.

8 Conclusion

An educational air blower prototype was developed, aimed to instrumentation and control laboratory teaching, but capable of being also used in fluid dynamics studies. It emphasizes some desired didactic characteristics, such as being playful, capable of emitting increasing noise when air speed rises, and provoking thermal sensation as temperature changes. It also "demonstrates theoretical ideas, reflects real-life problems, has a suitable timescale, is nonhazardous, inexpensive, easy to understand and use" (Balchen, 1981; Kheir, 1996). In fact, the air blower is a clean and safe process, needing little space at the workbench and also short times for the transients, thus spending less class time and proportioning more experimental tests to be done. Other prototypes have been developed following the same philosophy of this work, for instance, a reduced-scale elevator (Oliveira, 2014).

Two simple AC converters apply RMS controlled voltages to the DC series motor (for blowing) and to the resistor heater, highlighting a robust technique still in use by power electronics.

The blower automatic control requires air temperature and mass flow transducers, implemented with good accuracy and relatively simple electronic circuits. The first one is all analogical, while the second is half digital, half analogical, requiring also a software processing. This is an interesting application of signal electronics in instrumentation.

An Arduino board was used for real-time control. This hardware and software platform is largely diffused among students, and free software can be downloaded easily. From the PC side, a powerful and user-friendly Matlab human-machine interface permits the users to perform control in open-loop or in closed-loop, and also to visualize the controlled variables. In addition, the Matlab version has come with dedicated Arduino commands, permitting to program the system in a high level manner.

Black-box identification and model validation are easily done, and experimental results can be presented, showing the viability of the proposed reduced-scale process in teaching control practice.

Acknowledgments

We want to thank Depart. of Electronics Engineering/UFMG, CAPES/BRAFITEC Cooperation Agreement, Eng. André Martins Vaz and also the students who contributed to develop the first analog version of the Air Blower (2005-2008).

References

Aguirre, L.A. (2007). Introdução à Identificação de Sistemas. Cap. 3, 3ª ed., Ed. UFMG, Belo Horizonte, MG, Brasil.

Arduino. Board Mega 2560, https://www.arduino.cc/en/Main/arduinoBoardMega2560, access in 04/01/2015.

Arduino. IDE, v. 1.6.5, https://www.arduino.cc/, access in 04/01/2015.

Aström, K.J. and Hägglund, T. (2001). The future of PID control. Control Engineering Practice 9, Elsevier Science Ltd., pp. 1163-1175.

Balchen, J. G., Handlykken, M., Tysso, A. (1981). The need for better laboratory experiments in control engineering education. Control Science and Technology for the Progress of Society, Proc. of the Eighth Triennial World Congr. of IFAC, vol. 7, Pergamon Press, Kyoto, Japan.

Cortizo, P.C. (2006). Circuito para Comando de Gradadores, Depto. Eng. Eletrônica, EE/UFMG, Belo Horizonte, Brasil.

Fox, R.W. and McDonald, A.T. (2001). Introdução à Mecânica dos Fluidos, 5ª ed., LTC Editora, Rio de Janeiro, RJ.

INMET – Instituto Nacional de Meteorologia, http://www.inmet.gov.br/portal/index.php?r=estacoes/estacoesautomaticas, access in 11/01/2016.

Kheir, N. A. , Aström, K. J., Auslander, D., Cheok, K. C., Franklin, G. F., Masten, M. Rabins, M. (1996). Control Systems Engineering Education. IFAC Automatica. Great Britain, vol. 32, N. 2, pp. 147-166.

Lajnef, Y. (2015). Instrumentation and Control of an Air Blower, Stage Report, UFMG/ESIEE-Paris, 40 pp.

Matworks. Matlab R2015a, http://mathworks.com access in 04/01/2015.

Oliveira, J.C.R., Vaz, A.M. (2014). Teaching Control Lab with a Reduced-Scale Elevator, XIII Int. Conf. on Eng. and Tech. Education, Guimarães, Portugal, March16-19, pp. 68-72.

Sedra, A.S. and Smith, K.C. (1998). Microelectronic Circuits, chapter 11, Oxford University Press, New York.

Sonntag, R.E. and Borgnakke, C. (2003). Introdução à Termodinâmica para Engenharia, 1ª ed., LTC Editora, Rio de Janeiro, RJ.