Proceedings of the ASME 2019 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2019
August 18-21, 2019, Anaheim, CA, USA

IDETC2019-97664

LOW-COST REAL-TIME VISION PLATFORM FOR SPATIAL TEMPERATURE CONTROL RESEARCH EDUCATION DEVELOPMENTS

Jairo Viola
MESA Lab
School of Engineering
University of California, Merced
Merced, California, USA
Email: jviola@ucmerced.edu

Alberto Radici
MESA Lab
School of Engineering
University of California, Merced
Merced, California, USA
Email: a.radici001@studenti.unibs.it

Sina Dehghan
MESA Lab
School of Engineering
University of California, Merced
Merced, California, USA
Email: sdehghan@ucmerced.edu

YangQuan Chen *
MESA Lab
School of Engineering
University of California, Merced
Merced, California, USA
Email: ychen53@ucmerced.edu

ABSTRACT

Temperature control is present in many industrial processes, making this skill mandatory for the control engineers. For this reason, different training temperature platforms have been created for this purpose. However, many of these platforms are expensive, require elaborate facility accommodations, and have higher heating and cooling times, making not suitable for teaching and training. This paper presents a low-cost educational platform for temperature control training. The platform employs a Peltier module as a heating element, which has lower heating and cooling time than other thermal system implementations. A low-cost real-time thermal camera is employed as a temperature feedback sensor instead of a standard thermal sensor. The control algorithm is developed in Matlab-Simulink and employs an Arduino board as hardware in the loop to manage the Peltier module. A temperature control experiment is performed to show that the platform is suitable for teaching and training experiences not only in the classroom but for engineers in the industry.

*Corresponding author.

INTRODUCTION

Temperature is one of the primary process variables, which is an integral part of many industrial processes as oil refining [1], food production [2], agriculture [3], semiconductors manufacturing [4] among others. Thus, understanding the temperature dynamic behavior, as well as the different techniques for its control, is mandatory on the formation of the future control engineers. Although the nature of the temperature is studied in courses like thermodynamics or heat transfer, and the control techniques can be covered on a linear controls course, the control engineer needs to acquire practical skills to perform temperature control, which can be developed only on laboratory or training sessions, interacting with real temperature systems. So, there are different training rigs for temperature control that can be employed not only in academia but also for industry [5]- [9]. These platforms allow performing temperature control experiments employing industrial control elements as sensors, controllers or communication protocols giving to the future control engineer a closer approach to the industry temperature process control. However, these platforms are expensive with big sizes, making prohibitive for many engineering schools due to its cost and the required facilities accommodations. Besides, these platforms are not open hardware and software, limiting the possibility of implementing advanced control strategies further than Proportional Integral (PI) or Proportional Integral Derivative (PID) as well as testing new sensors or control elements. On the other hand, due to the slower dynamic of the thermal process and the size of the platforms, the heating and cooling cycles can take hours to be completed, making that each control experience takes a long time, extending laboratory and training sessions duration.

This paper presents a low-cost real-time temperature control platform for teaching and training purposes developed on the Mechatronics, Automation and Embedded systems laboratory (MESALab) of the University of California, Merced. The platform is open allowing the implementation of different control strategies and the integration of additional instrumentation. It is composed of a Peltier module, which is a low-cost heating element with shorter heating and cooling times compared with other thermal systems. The temperature feedback employs a low-cost FLIR thermal camera instead of thermocouples or RTD sensors, giving visual information about the dynamic behavior of the temperature on the Peltier module. The control algorithms are performed in Matlab-Simulink configured in hardware in the loop (HIL) with an Arduino board to send the control action to the Peltier module.

The main contribution of this paper is the development of a low-cost real-time vision temperature platform that can be employed on training and teaching experiences, allowing the implementation of different control strategies as well as it is easy transportation and installation. Indeed, the learning outcomes for the temperature training platform are in first place familiarize the students with the real behavior of a temperature system. On second place define a methodology for the temperature control problem following the steps of data acquisition, system identification, controller design, and implementation, and finally give to the engineering students the tools to take and solve a temperature control problem during their professional exercise.

The paper is structured as follows. Initially, the thermal system description is presented. Then, a temperature control experiment is performed with this platform showing its main features. Finally, conclusions and future works are presented.

TEMPERATURE TRAINING PLATFORM DESCRIPTION

The MESALab temperature training platform is presented on Fig.1. As can be observed, the system is composed by the Peltier module (P1), which is the heating element, the infrared thermal camera (P2) operating on a Raspberry Pi, and an Arduino board (P3) configured on HIL configuration with Matlab-Simulink to perform the data acquisition, identification and control tasks for the system. Besides, a power driver (P4) is included to manage the power applied to the Peltier module according to the control action defined by Matlab-Simulink control algorithm. A further description of the components is presented below.

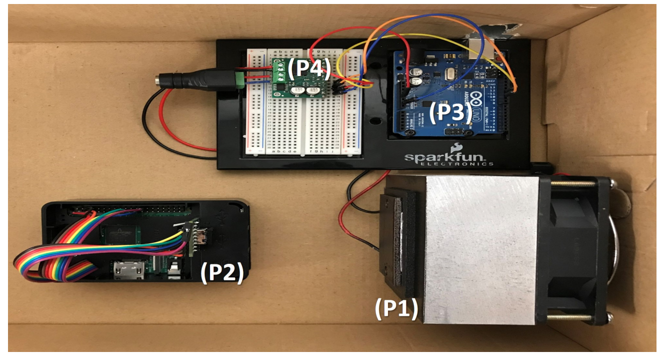


FIGURE 1: TEMPERATURE TRAINING SYSTEM

Peltier module

Peltier is a particular case of thermoelectric effect which produces a temperature change at an electrified junction of two different conductors. If an electric current across the junction, it generates a temperature change, which according to the direction of the current could produce a heating or cooling effect. The Peltier module shown in Fig.2 is employed as a heating element because it is a solid state device with low maintenance requirements and long service lifetime. The temperature range for this device comes from 15^oC to 50^oC, with a power requirement of $12v$, $5A$ delivering a maximum of $60W$. On the MESALab temperature platform, the power on the Peltier system is controlled using pulse width modulation (PWM) managed by the Arduino board and the power driver. The PWM signal range goes from -255 to 0 for cooling, and from 0 to 255 for heating, where -255 and 255 equals to the 100% of the PWM signal on cooling and heating operating modes. Also, the Peltier module includes a heat sink and an external fan to release the extra power produced on the back side of the Peltier plate to extend its lifetime.

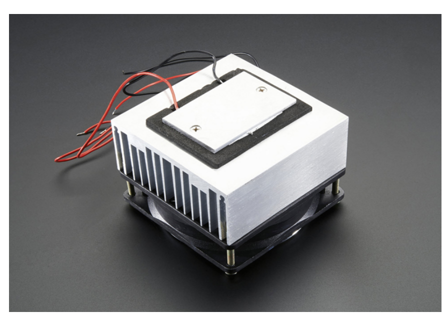


FIGURE 2: PELTIER MODULE

Infrared thermal camera

The thermal infrared camera employed as a temperature sensor for this system is shown in Fig.3. The camera is manufactured by FLIR, which is a long wave infrared camera that measures the temperature over a surface through its infrared emitted radiation [10]. The wavelength range for this camera comes from $8\mu m$ to $14\mu m$ with a maximum frame rate of 9 FPS. The camera has a resolution of 80x60 pixels with an accuracy of $\pm 0.5^o C$, and its size is less than a quarter coin. Besides, the camera has an SPI interface, which allows its connection with many edge devices. Also, the LeptonThread software development kit is available for the camera data acquisition, which runs in python and C++. For this platform, the thermal camera works together with a raspberry Pi 3B+, which reads the camera through the SPI interface, sending the data to Matlab-Simulink employing a TCP/IP client-server configuration. In this system, the thermal camera with the Raspberry Pi acts as the server, and Matlab-Simulink application runs as the client for the thermal data camera. An example of the infrared vision camera image working on the Raspberry Pi is presented on Fig.4, where the thermal camera detects the temperature changes on the Peltier plate and sends the temperature information using the TCP/IP configuration.

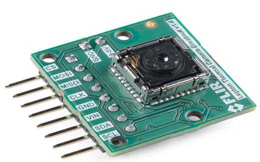


FIGURE 3: FLIR LEPTON THERMAL CAMERA

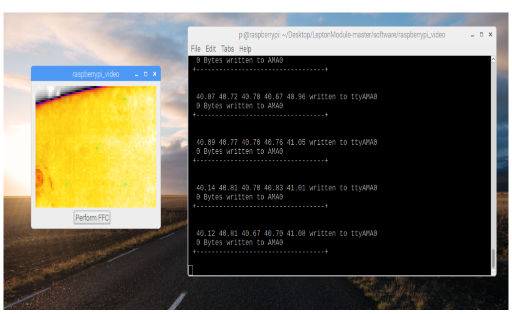
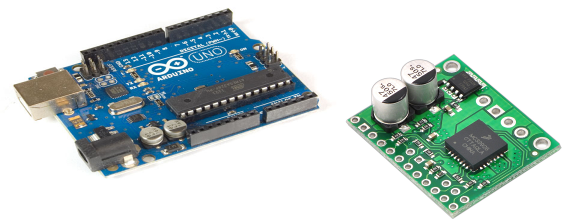


FIGURE 4: THERMAL IMAGE AND DATA INTERFACE

Arduino as HIL device

Fig.5a shows the Arduino UNO board used in this platform as a signal interface between the Peltier module and Matlab-Simulink control algorithm, which acts as an HIL device, transmitting the control action, given as a PWM signal from Matlab-Simulink to the temperature system. The usage of Arduino is motivated by its intuitive programming language and the different software tools for interacting with Matlab-Simulink. It is configured as HIL in order to obtain a real-time interaction of the system and Matlab-Simulink. On the other hand, Fig.5b shows the MC33926 DC motor driver employed as power interface to control the Peltier module using the Arduino. The driver works in a rage of $5V$ to $28V$ with a maximum current supply of $3A$ DC and peak current of $5A$ DC.



(a) (b)

FIGURE 5: ARDUINO UNO AND MC33926 MOTOR DRIVER

Matlab-Simulink interface

This training module employs Matlab-Simulink for the data acquisition, identification, and control tasks performed during the temperature control exercises. The purpose of employing Matlab-Simulink for this platform is the possibility of implementing many different control algorithms that go from the classic PI and PID controllers to robust, optimal, and nonlinear control strategies, using classic tools well known by most of the control engineers on academia or industry without the need of embedded implementation.

On the other hand, Matlab-Simulink allows interaction with different kind of sensors and control boards with different communication protocols. In this particular case, the infrared camera uses the TCP/IP communication protocol, and the Arduino board uses a serial communication protocol, both of them supported by Matlab-Simulink. So, adding new sensors and instruments to the system is possible without further complexity.

Total cost of the system

Table 1 shows the total cost of this platform corresponding to \$347.00 USD, which is an affordable cost for many engineering schools interested in temperature control training.

TABLE 1: TEMPERATURE TRAINING PLATFORM TOTAL COST

Component	Price
Peltier module	\$35.00
Raspberry PI 3B+	\$35.00
Arduino UNO board	\$20.00
Power driver	\$18.00
Thermal Camera	\$229.00
Others	\$10.00
Total cost	\$347.00

TEMPERATURE CONTROL TRAINING EXPERIMENT ON MESALAB PLATFORM

A training experiment is proposed in this paper to test the capabilities of the MESALab temperature platform for designing and implementing a visual temperature control. The experiment involves the stages of system identification, PID controller design using the Internal Model Control methodology (IMC), and its practical implementation using the HIL architecture.

Closed loop System configuration

Figure 6 shows the block diagram of the system configuration employed for the temperature training experiment. For this system, the Peltier module is the plant to be controlled, and the thermal infrared camera is used as a temperature feedback sensor. It communicates with Matlab-Simulink using TCP/IP protocol, where the system data is acquired and processed to produce the control action, which is sent to the Peltier module using the Arduino in HIL, with the power driver as the system actuator.

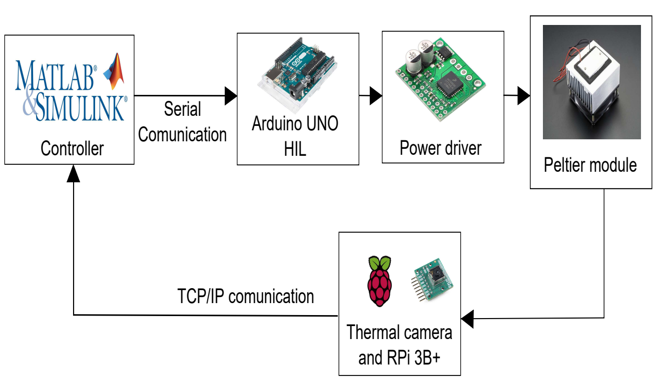


FIGURE 6: TEMPERATURE SYSTEM CLOSED LOOP CONFIGURATION

System identification

The dynamic behavior of the Peltier module is identified applying the stepped signal shown in Fig.7. This signal sets the maximum power to the Peltier module by $250s$, and then power off the module to analyze its natural cooling response. During the experiment, the environment temperature on the room is 22^oC. As can be appreciated, the heating and cooling processes for this system takes about 500 seconds (8 minutes). Figure 8 shows the Matlab-Simulink configuration employed to perform the data acquisition and the identification processes.

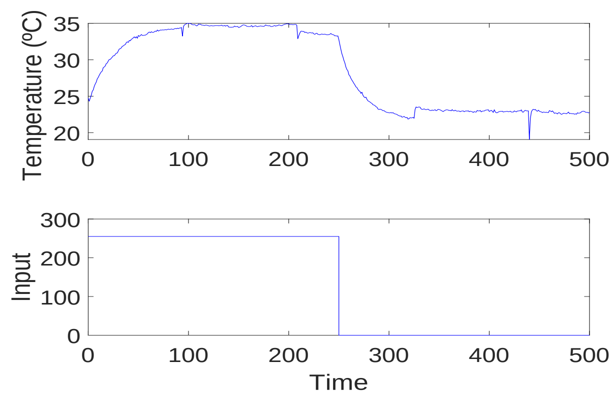


FIGURE 7: IDENTIFICATION SIGNAL AND PELTIER MODULE OPEN-LOOP TIME RESPONSE

$$G(s) = \frac{0.136}{(26.1s+1)(0.174s+1)}. \tag{1}$$

In Fig.8a, the TCP/IP communication block receives the temperature data from the Raspberry PI and the thermal camera. These data is related to five points in the peltier plate, which are located in the center line of the plate. From different measurements performed, the temperature in each point is similar. Therefore, one of these points is taken as the actual system temperature, which is presented in Fig.7. Likewise, the stepped input identification signal given by Fig.7 is sent to the Peltier system employing the Arduino UNO board as shown in Fig.8b. Employing the Matlab Identification toolbox, a second order model of the peltier system is obtained with a fitness of 90% given by (1), which time response vs the real data is presented in Fig.9.

Controller design

A PID controller (2) is designed for this system tuned with the internal model control technique (IMC). From [11], the PID controller proportional, integral, and derivative terms can be cal-

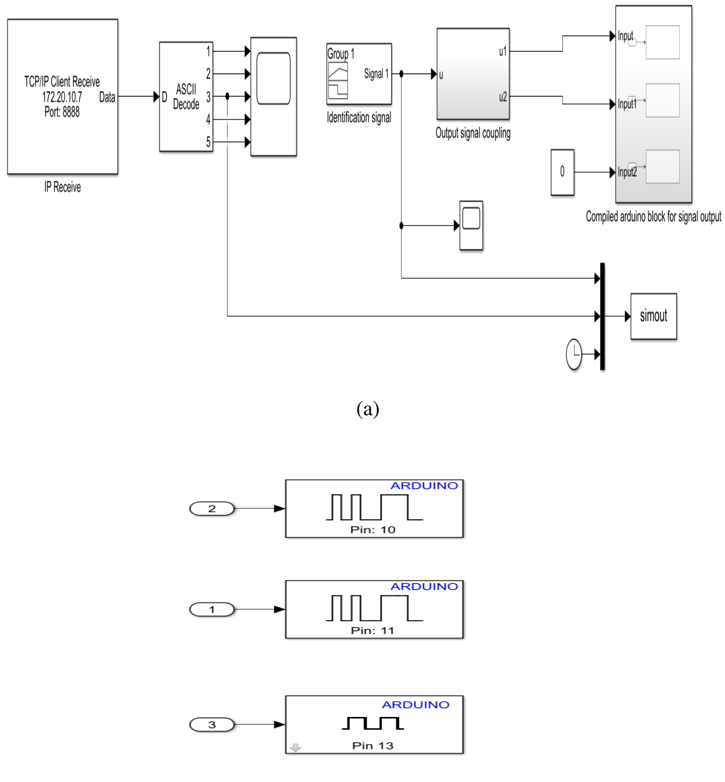


FIGURE 8: MATLAB-SIMULINK BLOCKS FOR ARDUINO INTERFACE

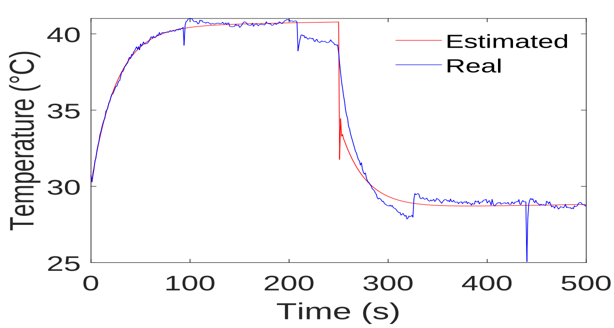


FIGURE 9: IDENTIFIED MODEL RESPONSE VS REAL DATA

culated for a second order with real poles system (3) using (4)-(6), where λ is the only tuning parameter that defines the closed loop speed response of the system. From (1), the terms of the PID controller are $k = 38.57$, $\tau_i = 26.26$, and $\tau_d = 0.1729$ with $\lambda = 5$ to obtain a smooth response of the system with no overshoot or $\zeta = 1$, a settling time of $100s$, and a rising time of $45s$.

$$PID(s) = k(1 + \frac{1}{\tau_i s} + \tau_d s) \tag{2}$$

$$G(s) = \frac{k_p}{(\tau_1 s + 1)(\tau_2 s + 1)}. \tag{3}$$

$$k = \frac{\tau_1 + \tau_2}{\lambda k_p} \tag{4}$$

$$\tau_i = (\tau_1 + \tau_2) \tag{5}$$

$$\tau_d = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2}. \tag{6}$$

Matlab-Simulink HIL controller implementation

The PID controller implementation is performed in Matlab-Simulink as shown in Fig.10. This configuration reads the temperature from the thermal camera and compares with the desired temperature of 30^oC, producing the error signal that feeds the PID controller, which generate the control action to be applied to the Peltier module using the Arduino in HIL configuration. According to [12], The sampling time choice T_s for a control system is given by (7), where τ is the time constant of the system.

$$T_s \leq \frac{\tau}{10}. \tag{7}$$

Calculating the time step response of (1), the steady state value of the system is reached at $t = 110s$. Therefore, τ is about four times less the steady state time, it means, $\tau = 27.5s$. Using (7),the sample time is $T_s \leq 2.75$. Considering that the thermal camera sample rate is $9Hz$ or 0.111, it is possible to say that $T_s = 0.111s$ is a good sampling time for the control system. Besides, it is important to notice that the PID controller block can be replaced by a different control strategy as an adaptive, model predictive controller or a nonlinear controller programmed with Matlab-Simulink in order to test an advanced control strategy without the need of embed implementation.

The system time response and the control action of the PID controller are presented in Fig.11. It can be seen that the system reaches the desired temperature setpoint of 30^oC with a smooth control action in about $100s$

Temperature platforms comparison

A comparison between some commercial temperature training rigs and the MESALab temperature platform is presented in Table 2 and Table 3. As can be notice in Table 2, the commercial rigs have bigger dimensions and require additional utilities as compressed air and current water. Also, the power consumption

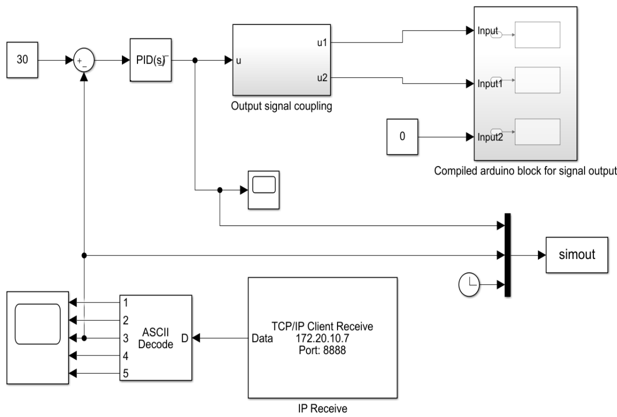


FIGURE 10: PID CONTROLLER HIL CONFIGURATION ON SIMULINK

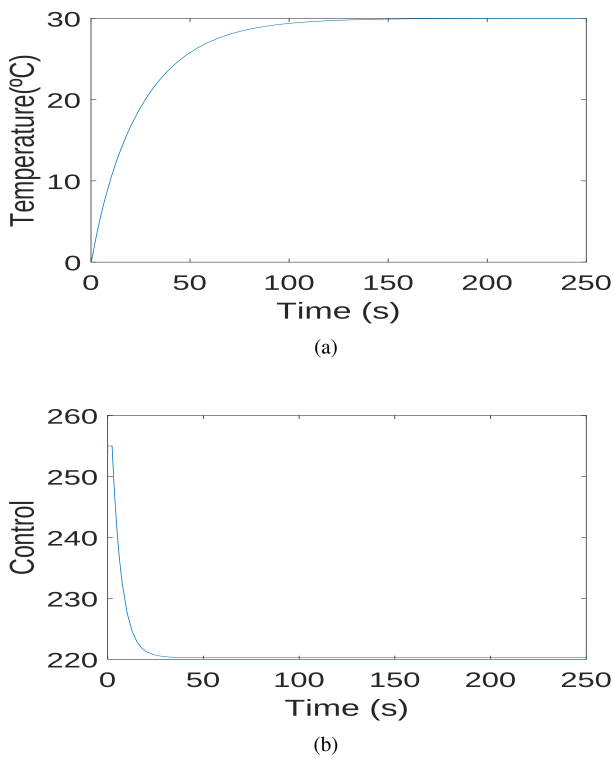


FIGURE 11: PID TEMPERATURE CONTROL A)TIME RESPONSE AND B)CONTROL ACTION

of these platforms is greater than $1kW$, which implies a higher energy consumption as well as higher heating and cooling times. On the other hand, the MESALab temperature platform has a smaller size than the commercial platforms enough to fit in a shoebox with minimum power consumption. Regarding to the temperature range, the MESALab platform range is not larger than the other platforms, but it is still adequate for different temperature control experiments. Moreover, the MESALab temperature platform does not need extra utilities as compressed air or three-phase electric power. From Table 3, it can be notice that the price of the MESALab temperature platform is significantly less expensive than the other commercial rigs, making affordable for many universities interested on teach temperature control in their undergraduate and graduate control courses. Besides, the heating and cooling cycle duration for the commercial platforms goes from 30 minutes to more than three hours, which is higher than MESALab temperature platform. For these reason, it is possible to say that the MESALab temperature platform is an alternative for temperature control training.

TABLE 2: TEMPERATURE TRAINING PLATFORMS FEATURES

Company	Temperature range (^oC)	Size (m) (LxHxW)	Power (W)
Festo MPS-PA Reactor	-40 to 200	1x0.8x1.8	1500
Tecquipment CE117	5 to 40	0.7x0.8x1.7	1500
Amatrol T5553	-20 to 145	1.6x0.3x0.5	1500
MESALab	15 to 50	0.3x0.2x0.1	70

CONCLUSIONS

This paper presented a low-cost temperature control platform for training future control engineers in thermal process control. The platform employs a Peltier module as a heating element, a low-cost infrared camera as a feedback sensor, and Matlab-Simulink with Arduino in HIL configuration for the data acquisition, identification, and control tasks. The overall cost of the platform is less than $\$350USD$ making affordable for many engineering schools. The temperature experiment performed on this platform shows that it is possible to develop different control techniques with a real-time response of the system. Besides, the future control engineers can follow a methodological procedure for the temperature control from the identification, controller design, and practical implementation, which can be employed for any thermal system. Moreover, the lower power consumption, as well as its small size and the only requirement of electricity for its working, makes the MESALab temperature platform suitable to perform training sessions inside and outside the classroom. Fi-

TABLE 3: TEMPERATURE TRAINING PLATFORMS PRICE AND HEATING-COOLING TIMES

Company	Price (USD)	Heating-Cooling time (minutes)
Festo MPS-PA Reactor	$21,025	200
Tecquipment CE117	$25,971	60
Amatrol T5553	$23,000	30
MESALab	$347.00	8

nally, the integration with Matlab-Simulink makes possible the use of advanced control techniques and its implementation on a real thermal system, expanding the research possibilities to developing new control techniques. For these reasons, we can conclude that the MESALab temperature platform is a system suitable for teaching thermal process control not only to engineering students but for people on the industry who need to acquire the skills to perform temperature control.

ACKNOWLEDGMENT

The authors are thankful to the members of PTUC/CPC (precision temperature uniformity control / cognitive process control) research group at UC Merced MESA Lab Jie Yuan for their contribution to this work.

REFERENCES

[1] Zhang S., Zhao D, Spurgeon S. K., and Yan X., "Distributed model predictive control for the atmospheric and vacuum distillation towers in a petroleum refining process," *2016 UKACC Int. Conf. Control. UKACC Control 2016*, pp. 1–6, 2016.

[2] Alvino A. , Matos M., Carpio C., and Vinces L., "Development of Automatic Control for Temperature-based Reduction of Bitterness in Tarwi," *2018 IEEE Int. Conf. Autom. Congr. Chil. Assoc. Autom. Control*, pp. 1–5, 2019.

[3] Dik D., Nikiforova T., Moskvin V., Polyakova E., and Chelovechkova A., "The System of Environment Control of Botanic Garden Greenhouses," *2018 Int. Multi-Conference Ind. Eng. Mod. Technol.*, pp. 1–7, 2019.

[4] J. Du, X. Zhang, and J. Shi, "A Condition Change Detection Method for Solar Conversion Efficiency in Solar Cell Manufacturing Processes," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 1, pp. 82–92, 2019.

[5] H. E. Corporation, "H-ICS-TT Temperature Process Loop, pp. 1–2, 2017.

[6] Tecquipment, "Temperature Process Training System," p. 2, 2016.

[7] Festo Didactic Inc., "MPS-PA workbook." p. 29, 2007.

[8] Amatrol. Inc., Temperature process control training. 2015.

[9] C. Uyanik, M. Özkan, and O. Parlaktuna, "Design of an experimental platform for process control systems," *ELECO 9th Int. Conf. Electr. Electron. Eng.*, pp. 812–816, 2016.

[10] F. systems Inc, "FLIR LEPTON Infrared camera," last modified February 19, 2019, accessed May 24, 2019: https://goo.gl/RnKbKS

[11] D. E. Rivera Flores, "Una metodología para la identificación integrada con el diseño de controladores imc-pid," *Rev. Iberoam. Automática e Informática Ind. RIAI*, vol. 4, no. 4, 2007 pp. 5–18.

[12] A. Visioli, *Practical PID Control-(Advances in industrial control)*, Springer, Vol. 1. 2006.