Can Tho University Journal of Science
website: ctujs.ctu.edu.vn

DOI: 10.22144/ctu.jen.2022.007

Application of different control algorithms on a 'home-made' temperature control lab kit

Nguyen Nhu Tuong An[1], Tran Nguyen Tuan[2], Pham The Thinh[1] and Do Vinh Quang[1*]

[1]*Faculty of Mechanical Engineering, Can Tho University of Technology, Viet Nam*

[2]*Student of Automation & Control Engineering, Can Tho University of Technology, Viet Nam*

[*]*Correspondence: Do Vinh Quang (email: dvquang@ctuet.edu.vn)*

Article info.

Received 30 Aug 2021
Revised 05 Nov 2021
Accepted 05 Nov 2021

Keywords

Arduino, control algorithms, MATLAB, temperature control lab kit, system modelling, Simulink

ABSTRACT

Providing enough facilities for students to do laboratory activities is important. An existing useful kit was proposed for students learning a variety of control engineering topics. A temperature control lab kit is made from scratch using common electronics components as a replacement for the original TCLab introduced by Hedengren (Hedengren et al., 2019). Mathematical models of the system derived from theoretical and experimental methods are simulated in Matlab/Simulink to verify their accuracy to the physical kit. Different control algorithms such as: On/Off, PID, Fuzzy are then applied on the Kit as well as its mathematical models to illustrate their control feasibility. Human machine interface (HMI) is also designed using Matlab GUI allowing an operator to select a control algorithm, tune control parameters and monitor parameters of the process. Experimental results show that the derived models can reflect quite well dynamics of the physical kit with temperature deviation among them in the range of ±3°C. This confirms that the kit is well-suited for teaching different control topics such as system modelling, system identification, classical control and advanced control algorithms.

1. INTRODUCTION

Laboratory activities are important, especially in the field of engineering. It not only provides engineering academics a better understanding of theoretical issues via their applications in real scenarios but also an appreciation of the imperfect match between theory and real application (Rossiter, 2017). There have been different kinds of facilities (including hardware and software) providing such activities. (a) Traditional laboratories with facilities provided from professional suppliers such as SMC (MAP-200, AUTOMATE-200, FMS-200), Festo (TP-240, MPS system 403-1), Siemens (SCE), National Instrument (mechatronics lab, temperature measurement trainer), etc. Those facilities are robust, high accuracy, and similar to industrial equipment. However, they are expensive; students have limited access time and require a lab supervisor to be allowed to perform an activity. (b) One of the solutions to this limitation is to use virtual laboratories (including web-based laboratories) where space and accessing time are not restricted (Rossiter, 2017; Goodwin et al., 2011). However, they are models; they are not real and teachers want their students to work with hardware (Rossiter et al., 2019). (c) Therefore, some universities provide "take-home equipment" in which students can take equipment home to perform experiments (Rossiter et al., 2019). (d) In order to encourage bachelor students to complete a product, based on the idea of academic group projects, students are required to do "home-made equipment". By doing this, other skills apart from system modeling, applying control algorithms could also be obtained such as reading datasheets of electronics components, designing and making a Printing Circuit Board (PCB), soldering, and testing.

Different laboratory facilities have been considered which should be suitable to our teaching objectives such as control engineering-oriented, compact, easy to make and use, and low price. Equipment for temperature monitoring and control is a good choice because temperature is one of the very important physical quantities that its manipulation is indispensable in many industrial sectors. On laboratory scale, there are different temperature control kits available and some can be conveniently interacted to Matlab/Simulink environment. Singhala (Singhala et al., 2014) designed a 'home-made' low-cost temperature controller embedded Fuzzy algorithm. The microcontroller generates a PWM (Pulse Width Modulation) signal for the MOSFET to allow a desired amount of power to fan and heater. This structure is suitable for students to implement but does not mention supporting interaction to Matlab/Simulink. aMG Temperature Control LAB KIT (Aimagin, 2021) is a commercial product based on FiO boards (ARM Cortex - M architecture) and RapidSTM32 Blockset (a 3rd party Simulink add-on Blockset). Temperature is controlled by 2 fans and 1 heater (halogen lamp). The kit is professional with many functions however it is over 100 USD and does not support the first principle of modeling. The Control tutorials for Matlab & Simulink website offers a tutorial for temperature control of a lightbulb (Simulink, 2021). The structure is straightforward for students to make and verify; however, second-year students are not recommended to work with AC current alone; in addition, Relays are slow-switching component to realize PWM. By further simplifying the heater to be a power transistor, a fan into a heat sink, the TCLab, developed at Brigham Young University (Hedengren et al., 2019) at a price of 35 USD, is a good candidate for "home-made equipment". In fact, the kit has been used for teaching control (de Moura Oliveira et al., 2020) chemistry and computing (Park et al., 2020) at some universities, and it is currently being used widely. Furthermore, Prof. John D. Hedengren and his team have launched a website in which the TCLab is intensively applied to demonstrate different techniques related to Process Control Engineering (https://apm.byu.edu/prism/index.php/Site/OnlineCourses). Unfortunately, this TCLab kit has not yet been available in Vietnam and is still rather costly for many of our students. With the purpose of popularizing this useful kit to students, this study has proved that the kit can be made from scratch with similar components that are cheap and easy to find and that the "home-made" kit is eligible for use in teaching and learning control-related topics by comparing the dynamic response of the physical model to its mathematical models under different control regimes.

2. SYSTEM CONSTRUCTION

2.1. System Overview

The "home-made" temperature control lab kit ("home-made" TCLab kit), shown in Fig. 1d., is a twin system to implement two separate control loops (one for temperature control and another for creating temperature disturbance). This physical kit is built from components that are cheap and easy for students to buy (Table 1) following the procedure shown in Fig. 1.

Table 1. Components in the "home-made" TCLab kit

No.	Part Type	Amount	Description
1	Arduino Uno	1	Micro-controller board for hardware and software communication, data acquisition, and analog output control (PWM 0 – 100%)
2	LED	1	Indicator for power supply to the heater (Tip41C)
3	Tip41C	2	Heater (withstand temperature up to 150°C)
4	LM35	2	Temperature sensors (measuring range: -55°C to 150°C)
5	1kΩ Resistors	3	Tolerance ±1%, resistance 1kΩ for limiting current
6	Power Supply	1	Provides power to transistor heaters (5V – 2A)
7	Heat sinks	2	Connecting to Tip41C for heat dissipation

Fig. 1a represents the system's working principle in which temperature of the heater is constantly measured by a sensor and sent to the computer (PC) via the Arduino UNO. A control algorithm, which is implemented in Matlab/Simulink software, uses the measured temperature to calculate a control signal that manipulates the heater to a desired temperature. This closed-loop control structure can be realized by connecting hardware components as in Fig. 1b. But for a professional outlook as a shield of the Arduino board, the schematic diagram of the circuit is converted into a PCB single layout using Proteus software. The complete system is shown in Fig. 1c. With the Simulink® Support Package for Arduino® Hardware (Mathworks, 2021) or Simulink ArduinoIO Package (https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_IOpack), users can interact with the system from Matlab/Simulink.

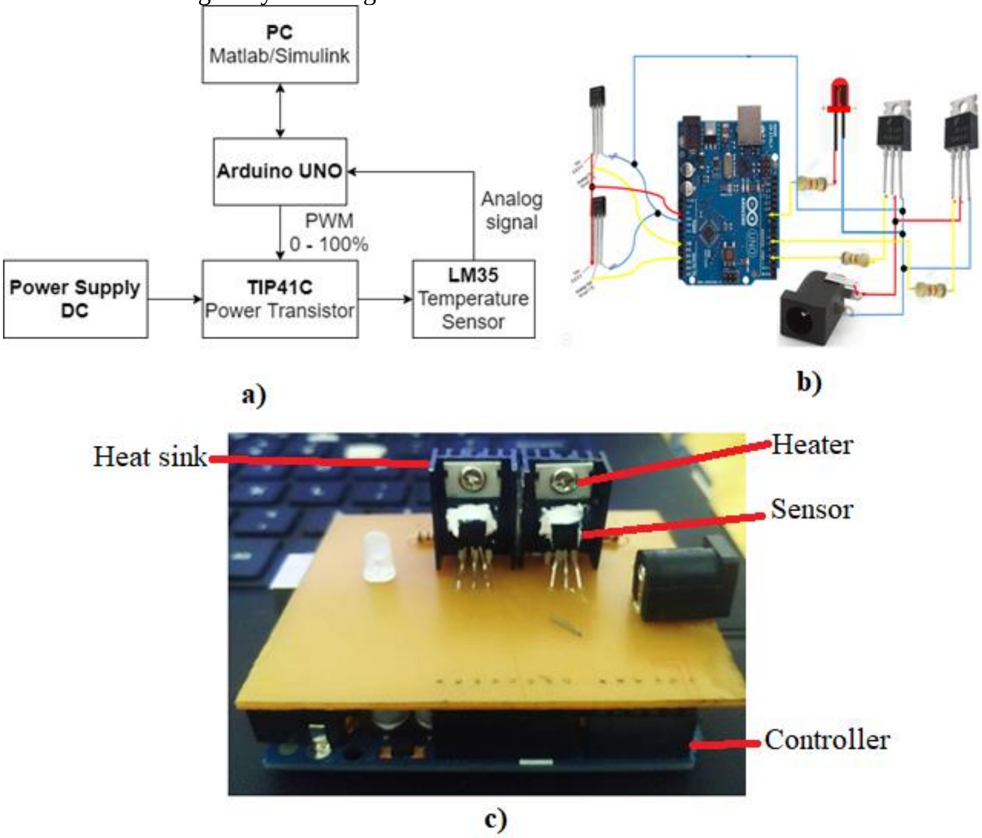


Figure 1. Overall building procedure for the "home-made" TCLab kit

(a) System diagram, (b) Circuit diagram, (c) The completed kit

2.2. System modeling

Representing dynamics of a physical system by mathematical equations is an indispensable step in the procedure of system analysis and controller design. Behaviors of the system subjected to different inputs are properly described via process variables that can be graphically visualized or dynamically simulated through various software such as MATLAB, LABVIEW. As a consequence, the system's dynamics can be better understood and the controller can be designed effectively (Ogunnaike & Ray, 1994). The following subsections are about the two methods commonly described in an academic textbook for obtaining a math model as well as their application to the "home-made" TCLab kit. The math model derivation of the kit is also available in apmonitor.com (John D. Hedengren) in different lecture notes.

2.3. System identification

From the physical kit (or physical model) that has been fabricated above, a mathematical model can be derived from measured input/output data. The program for obtaining data is represented in Fig. 2. By supplying 100% power to the TIP41C heater via the constant function block for 300 seconds, its open-loop response is recorded as shown in Fig. 3. Applying the tangent line method to this graph, the

FOPDT (first order plus dead time) model of the kit representing the relation between the input power (input) and the temperature response of the heater (output) is obtained as in (1).

$$G(s) = \frac{K}{\tau s+1} e^{-\theta s} \quad (1)$$

Where K = 52 is the process gain, $\tau = 120$ is the processing time constant and $\theta = 4$ is the process time delay.

By approximating $e^{-\theta s} \approx \frac{1}{1+\theta s}$, the transfer function (1) becomes:

$$G(s) = \frac{52}{120s+1} \times \frac{1}{4s+1}$$

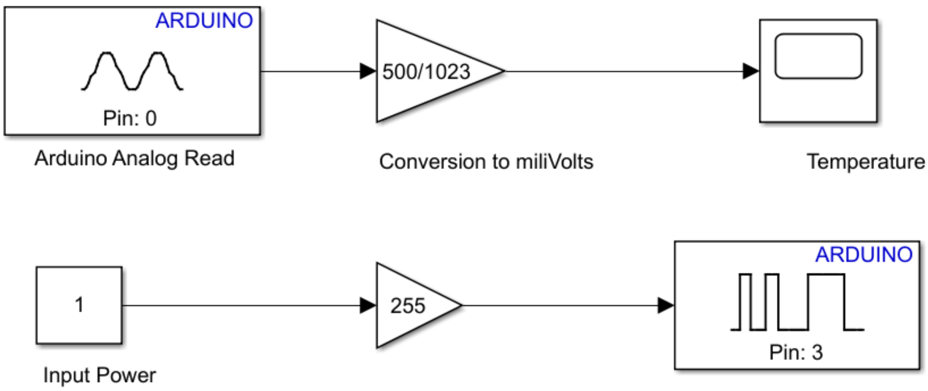


Figure 2. Program in Matlab/Simulink for powering the heater and reading its data

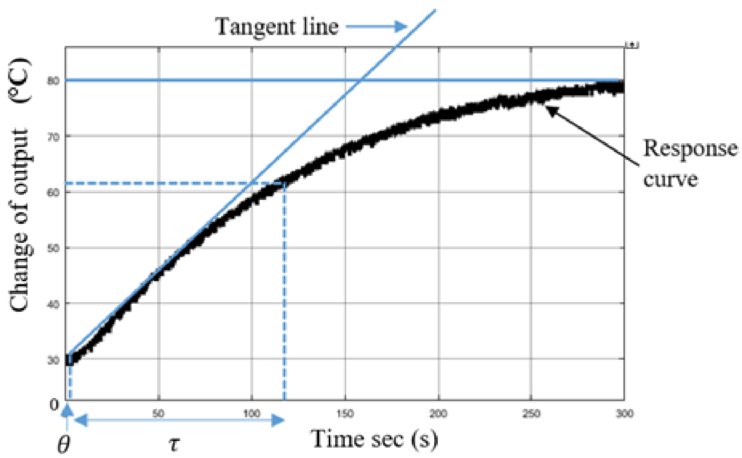


Figure 3. Tangent method on transient response graph

2.4. System modeling from the first principle

The dynamic model of the temperature system can be found using the principle of energy conservation. The rate of energy accumulation of the heater is the summation of the rate of input energy from the power supply and the rate of energy loss to the surrounding via the convection and radiation process (https://apmonitor.com/pdc/index.php/Main/TCLab Radiative).

$$mCp\frac{dT}{dt} = UA(T_a - T) + \varepsilon\sigma A(T_s^4 - T^4) + \alpha Q \quad (3)$$

Where:

T: °C (heater's temperature)

T_a: °C (ambient temperature)

T_s: °C (Temperature of surrounding objects, $T_s = T_a$ in this case)

m= 0.004 kg (mass of the heater)

C_p = 500 J/kg-K (specific heat capacity of the heater)

A = 0.0012 m^2 (area of the heat sink)

U = 5 W/m^2-K (convective heat transfer coefficient of free air)

$\varepsilon = 0.9$ (emissivity of the heater)

$\sigma = 5.67\text{x}10^{-8}$ W/m^2-K^4 (Setfan–Boltzmann constant)

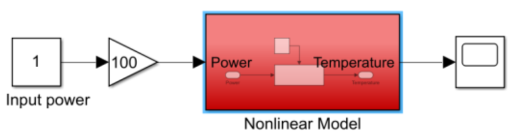
$\alpha = 0.01$ W/% (value of α depends on the input power and the heater)

Q: % (Percentage of input power)

Therefore,

$$\frac{dT}{dt} = f(T, Q) = \frac{UA(T_a - T) + \varepsilon\sigma A(T_a^4 - T^4) + \alpha Q}{mC_p} \quad (4)$$

This nonlinear equation is also known as the differential equation of the physical kit that relates the input power and the temperature output. The model can be simulated in Matlab/Simulink as in Fig. 4 in which the "Nonlinear Model" function block includes a sub-block "Triggered Subsystem" (Fig. 4b) that is executed conditionally via the triggered signal. The output signal of "Triggered Subsystem" depends on the parameters and algorithms inside the "Interpreted MATLAB Fcn" as shown in Fig. 4c and Fig. 4d.

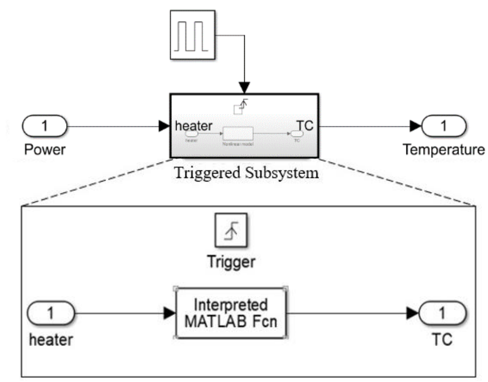


a)

```
function dTdt = thong_so(time,T,heater)
m = 4/1000;
U = 10;
A = 12/100^2;
alpha = 0.01;
Cp = 500.0;
Ta = 30+273.15; %K
sigma = 5.67e-8;
eps = 0.9;

dTdt = (U*A*(Ta-T)+sigma*eps*A*(Ta^4-T^4)+alpha*heater)/(m*Cp)
end
```

c)



c)

```
function TC = fcn_fcn(heater)
persistent T0 icount
if (isempty(icount)),
    T0 = 30 +273.15;
    icount = 0;
end
icount = icount+1;

time = [0,1];
[time,T] = ode15s(@(t,x)thong_so(t,x,heater),time,T0);
TK = T(end);
T0 = TK;
noise =(rand()-0.5)*2.0;
TC = TK - 273.15 + noise;
end
```

d)

Figure 4. (a) Simulation program of nonlinear model in Matlab/Simulink, (b) Function block of "Nonlinear Model", (c) Declaration of parameters for the differential equation, (d) Solving the differential equation

Simulation results for the experimentally derived linear model (blue line), theoretically derived nonlinear model (black line) in comparison to the actual result obtained from the physical kit (red line) in response to an input signal stepping from 0 to 100% are shown in Fig. 5. Simulation period is 300 seconds with initial temperature equal to the ambient temperature of 30°C. The nonlinear model is added a random noise ±2°C in order to imitate the influence of the ambient environment.

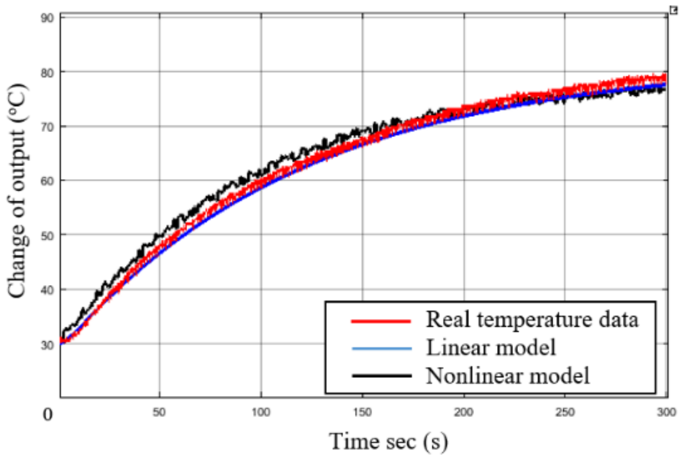


Figure 5. Step responses of the linear model, nonlinear model and physical model

As it can be seen, the temperature of the above models increases from 30°C to about 77°C with a variation within ±3°C. This means that the linear model and nonlinear model are good representations of the physical kit in the range from 0 to 100% of the input signal. The real data has noise as expected which is caused by the external environment as well as measuring noise. In conclusion, these models can be used for testing different control algorithms before embedding the suitable one to the physical kit.

3. APPLICATION OF CONTROL ALGORITHMS

The control of temperature for the kit can be performed using the closed-loop control structure (Fig. 6.). A signal of the actual output (T_p) is feedback to compare with the desired output (a reference or setpoint, T_{ref}) to give out the error (*e*). This error signal is then used by different control algorithms such as (ON/OFF, PID, FUZZY, etc.) to determine the control signal (*U*) that impacts on the plant. The following subsections show the implementation of some control algorithms to the kit.

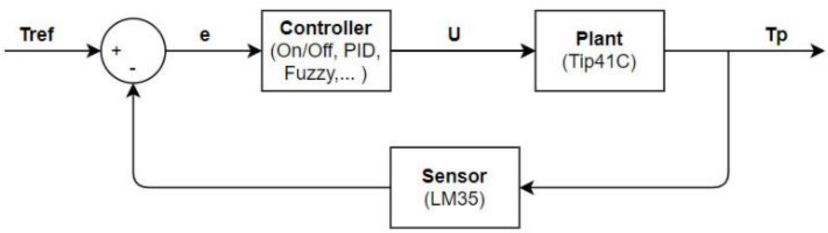


Figure 6. The block diagram of the closed-loop control system

3.1. On/Off control

In this algorithm, if the error (*e*) is positive, the controller will fully turn on the heater (100% power supply) to maintain the setpoint temperature. On the contrary, if the error (e) is negative, the controller will turn off the heater (0% power supply). While the heater turns off and on, it can be maintained at the desired temperature. The control and simulation program in Simulink for the three models is shown in Fig. 7. The program is simulated for 300s with a setpoint of 50°C starting at room temperature (approx. 30°C). The results are plotted in Fig. 8 with 4 lines for setpoint temperature (pink), real measured temperature (red), simulated temperature from the linear model (blue), and simulated temperature from the nonlinear model (black).

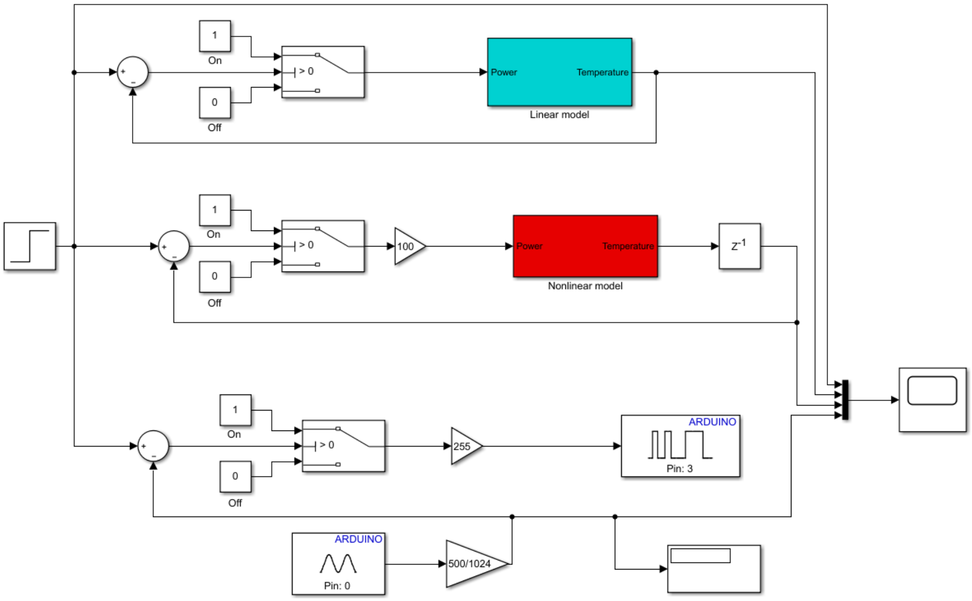


Figure 7. Simulation program of three models with On/Off controller

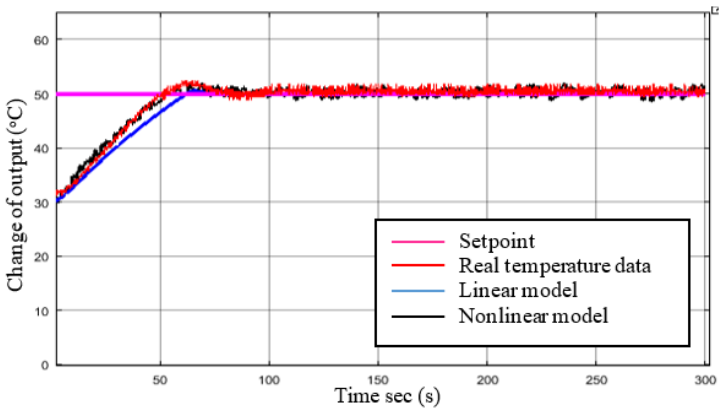


Figure 8. On/Off control for various models of the kit

Simulated results are approximately in accordance with the real data measured from the heater. They all reach and stay at the setpoint of 50°C with the rise time of about 50 seconds and a settling time of about 70 seconds. The real temperature (red line) has the highest overshoot of all (about 10%) due to the inertia of the heating process. The On/Off control algorithm works well in this application; however, its instinctive drawback is the possibly high switching frequency which remarkably reduces the lifetime of the actuator (specifically in mechanical-related actuators).

3.2. PID control

PID controller has been the most widely used control technique for years (Chen & Seborg, 2002). There are numerous techniques for PID design, tuning, and application examples that can be found in literature easily (Chen & Seborg, 2002; Wu et al., 2014). In this research, the Internal Model Control (IMC) (Francis & Wonham, 1976) tuning method is employed. The PID tuning parameters are calculated first using the FOPDT (first order plus deadtime) model. Those are then applied and simulated in the linear and non-linear mathematical models for possible tuning. After the closed-loop system achieves the desired performance, these tuned values are deployed to the real system.

Detailed calculation for tuning parameters of the Dependent, ideal PID controller form (5) is shown in table 2. This calculation applies for (1) with the process gain $K = 52$, process time constant $\tau = 120$, process time delay $\theta = 4$, and closed-loop time constant T_c = 12 (in this selection of moderate response, T_c is the larger between $0.1 * \tau$ and $0.8*\theta$) (Arbogast et al., 2010; Guru, 2015).

$$\frac{U(s)}{E(s)} = K_c\left(1 + \frac{1}{Ti}\frac{1}{s} + T_d s\right) \quad (5)$$

Table 2. IMC method for calculation of PID parameters

Controller \ Parameters	K_c	T_i	T_d
P	$\frac{0.2}{K}\left(\frac{\tau}{\theta}\right)^{1.22} \approx 0.24$		
PI	$\frac{1}{K}\left(\frac{\tau}{\theta + T_c}\right) \approx 0.14$	τ =120	
PID	$\frac{1}{K}\left(\frac{\tau + 0.5\theta}{0.5\theta + T_c}\right) \approx 0.17$	$\tau + 0.5\theta = 122$	$\frac{\tau * \theta}{\tau + \theta} \approx 3.87$

Matlab/Simulink programs for the three models are shown in Fig. 9. After some fine-tuning, the program is simulated for 300s with the settings for testing the ON/OFF controller as well as the legends for the plotted results (Fig. 10).

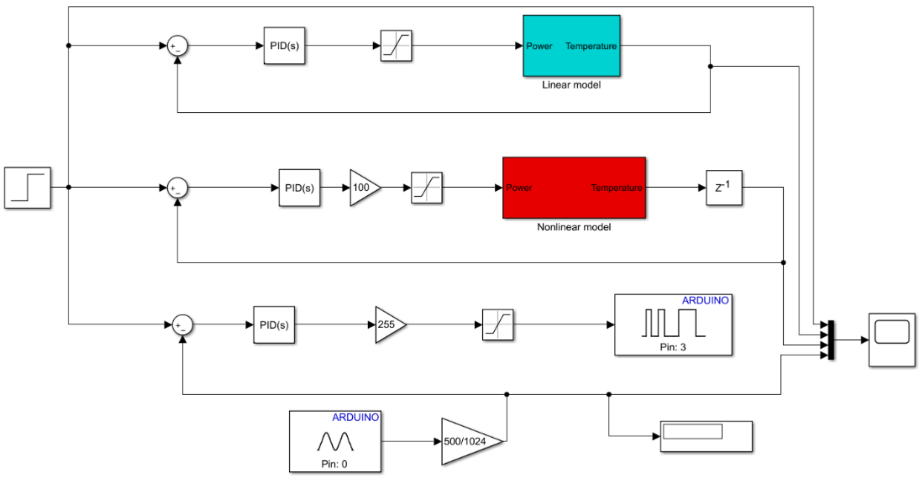


Figure 9. Simulation program with IMC-based PID controller

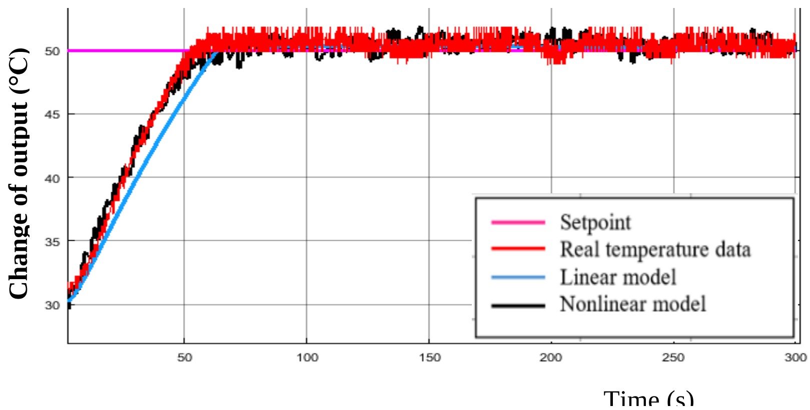


Figure 10. IMC-based PID control for various models of the kit

Again, output responses are similar for all models as the red, blue and black lines are close together, which means that the derived mathematical models reflect properly the dynamics of the physical kit. They settle to the setpoint (50°C), with the rise time about 50 seconds and settling time of about 60 seconds, which is approximate to that of the ON/OFF algorithm. However, in the PID algorithm, with good sets of tuning parameters, the response can quickly reach the steady state without overshoot.

3.3. Fuzzy control

Fuzzy logic was first introduced in 1965 by Lotfi Zadeh at University of California, Berkeley (Broesch, 2009; Passino et al., 1998). It is one of the most popular intelligent control techniques that convert the linguistic control strategy into an automation control strategy based on expert knowledge and fuzzy logic (Elnour & Taha, 2013). In this research, a simple fuzzy controller (Mamdani type, centroid defuzzification method) with two inputs and one output is applied to further demonstrate the utility of the kit.

As plotted in Fig. 5, the heating range of the system is approximately from 30°C to 80°C. Therefore, the maximum positive error is 50°C. However, as T_{ref} is not set higher than 50°C and the input range is free to choose, the selection for the error, *e*, and its derivative, *de*, is *e* = [-5; 27], *de* = [-5; 5] respectively and the output *U* = [0;1] since the heater can heat up only. After some trials and errors of testing and tuning, the selections of membership functions and rules matrix are showed in Fig. 11 and in Table 3 respectively.

Table 3. Rules matrix

U		de		
		Down	*Flat*	*Up*
e	*Negative*	Small	Small	Small
	Zero	Medium	Large	Large
	Positive	Large	Large	Large

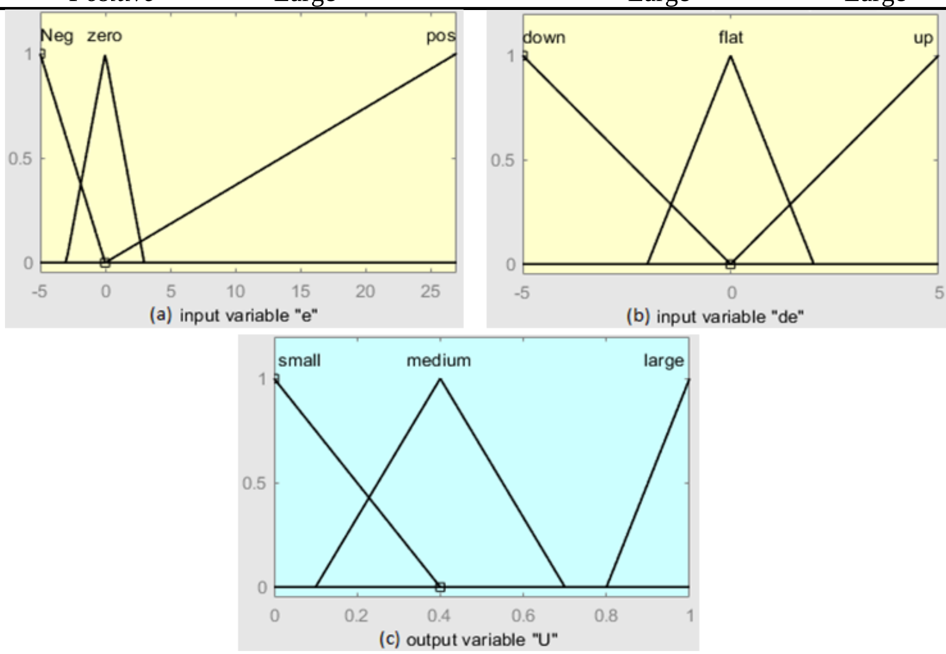


Figure 11. Membership functions: (a) Error (e), (b) Rate of change of error (de), (c) Control output (U)

Again, the three models are tested using a Matlab/Simulink programs as shown in Fig. 12. The program is simulated for 300s with temperature setpoints as well as the legends for the plotted results (Fig. 13) for testing the ON/OFF controller.

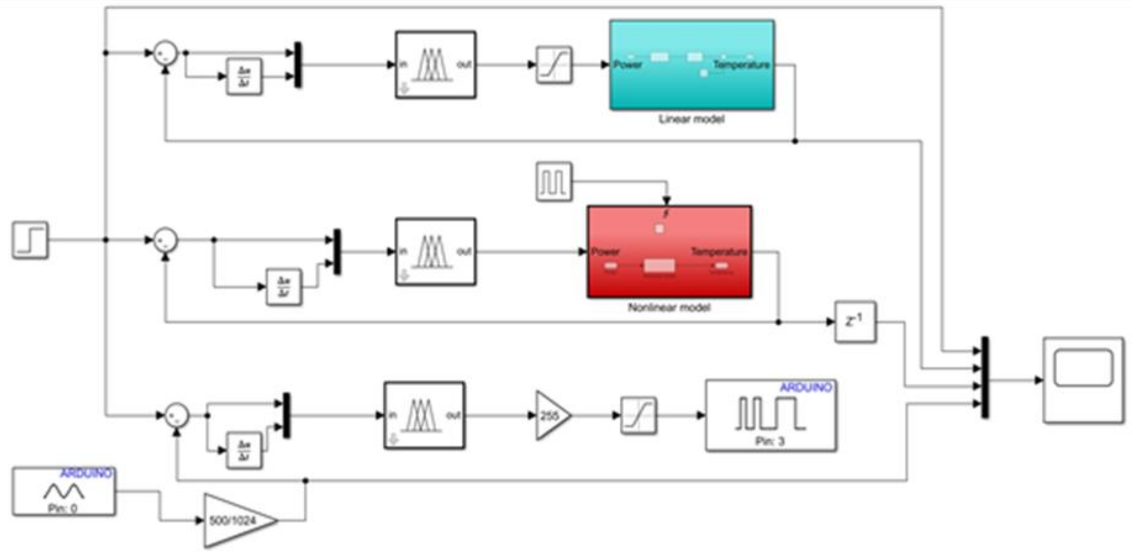


Figure 12. Simulation program with Fuzzy controller

As expected, responses for all models are overlapped since it has been proved that the linear and nonlinear models reflect properly the dynamics of the physical kit. They all settle to the setpoint (50°C) with rise time of about 50 seconds, settling time of about 65 seconds, and negligible overshoot.

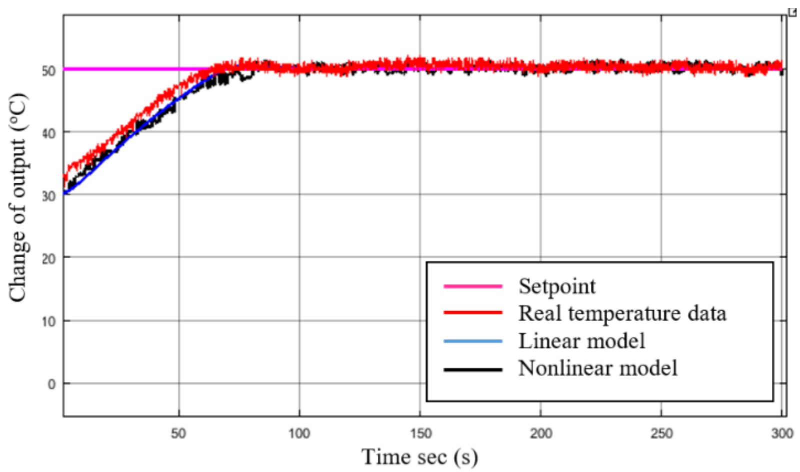


Figure 13. Fuzzy control for various models of the kit

3.4. Disturbance rejection in the control algorithms

A physical system is always subjected to disturbances, for example, interference from the surrounding environment, measurement noise, uncertainties, nonlinearities, time delays which degrade system performance (Wei, 2018) . Therefore, a good selection of the tuning parameters must help the closed-loop system robust to these disturbances. In this kit, the second TIP41C is targeted as the main source of external interference (the disturbing heater). To all three controllers used, the disturbing heater is provided with constant power (20%) for heating whose temperature will directly affect the system's output and to the sensor LM35.

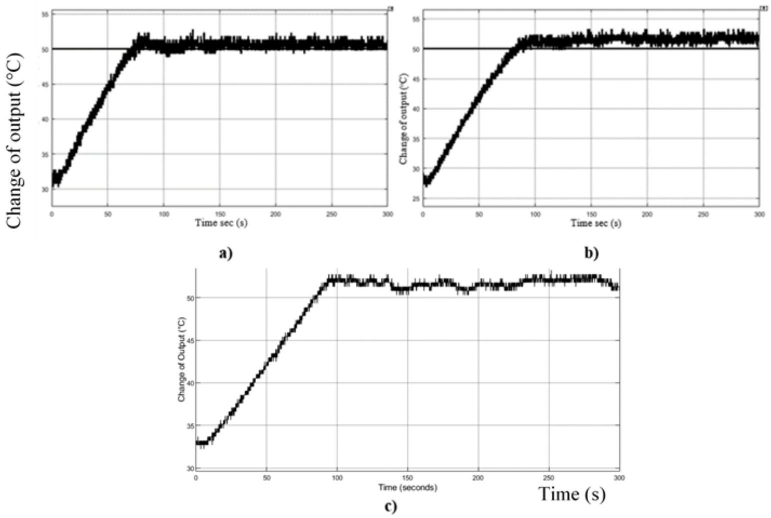


Figure 14. Disturbance rejection of control algorithms: (a) On/Off, (b) PID, (c) Fuzzy

Fig. 14 shows the closed-loop response of the kit under the interference when implementing three controllers respectively: (a) On/Off, (b) PID and (c) Fuzzy controller. In general, the system remains stable under the manipulation of those controllers. The interference causes a deviation at a steady state of about 2°C in PID controller, 2.5°C in Fuzzy controller, and is negligible in ON/OFF control. The reason could be the high switching frequency of ON/OFF control and fast dynamics of the heating process that the electrical current influences immediately on the temperature of the heater (TIP41C). Meanwhile, for PID controller, the integral action with high integral time is rather slow and the derivative action worsens the system performance due to measurement noise.

4. SCADA

For completeness, a Graphical User Interface (GUI) of the "home-made" TCLab kit is designed via the App Designer in MATLAB. In which, the visual components in Fig. 15 interact with Simulink components controlling the hardware via callback functions. The interface allows users to perform opened-loop control like manual switching of the heater or closed-loop control like real-time monitoring of temperature. For every change of the GUI's parameters such as setpoint, selection of the control algorithms or setting of control parameters, users have to Switch On again. For an undisrupted Supervisory Control and Data Acquisition (SCADA), LABVIEW is a good candidate.

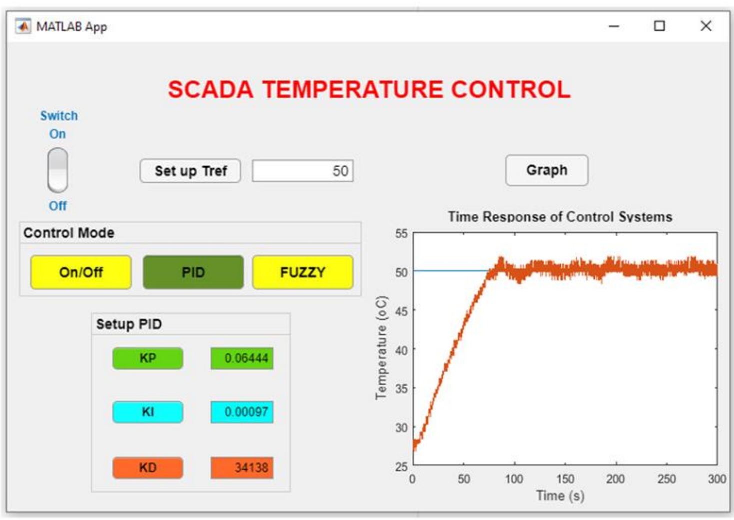


Figure 15. HMI in Matlab GUI

5. CONCLUSIONS

A process of making and testing a low-cost temperature control lab kit using power transistor and temperature sensor as a replacement for the original TCLab are presented. The physical model is compared to the two mathematical models derived from the first principle method and the system identification method, whose results show the similarity in their dynamic behaviors. By this agreement, different control algorithms such as: On/Off, PID, Fuzzy are tested on the math models first in order to achieve a suitable set of tuning parameters before implementing them on the physical kit. All the results solidify that the kit can be made by undergraduate students and used as lab equipment for them to learn various topics in control engineering such as system modeling, system identification, classical control, advanced control algorithms as well as system's robustness.

REFERENCES

Aimagin. (2021). How to Read Thermocouples. Retrieved from https://support.aimagin.com/projects/support/wiki/How_to_Read_Thermocouples

Arbogast, J. E., Cooper, D. J., & Rice, R. C. (2010). Model-Based Tuning Methods for PID Controllers.

Broesch, J. D. (2009). Chapter 7 - Applications of DSP. In J. D. Broesch (Ed.), Digital Signal Processing (pp. 125-134). Burlington: Newnes.

Chen, D., & Seborg, D. E. (2002). PI/PID Controller Design Based on Direct Synthesis and Disturbance Rejection. Industrial & Engineering Chemistry Research, 41(19), 4807-4822. doi:10.1021/ie010756m

de Moura Oliveira, P., Hedengren, J. D., & Rossiter, J. (2020). Introducing Digital Controllers to Undergraduate Students using the TCLab Arduino Kit. IFAC-PapersOnLine, 53(2), 17524-17529.

Elnour, M., & Taha, W. I. M. (2013, 26-28 Aug. 2013). PID and fuzzy logic in temperature control system. Paper presented at the 2013 International Conference On Computing, Electrical And Electronic Engineering (ICCEEE).

Francis, B. A., & Wonham, W. M. (1976). The internal model principle of control theory. *Automatica, 12*(5), 457-465. doi:https://doi.org/10.1016/0005-1098(76)90006-6

Guru, C. (2015). PID control of the heat exchanger. https://controlguru.com/pid-control-of-the-heat-exchanger/

Hedengren, J. D., Martin, R. A., Kantor, J. C., & Reuel, N. (2019). Temperature Control Lab for Dynamics and Control. Paper presented at the 2019 AIChE Annual Meeting.

Mathworks. (2021). Install Support for Arduino Hardware. https://www.mathworks.com/help/supportpkg/arduino/ug/install-support-for-arduino-hardware.html

Park, J., Martin, R. A., Kelly, J. D., & Hedengren, J. D. (2020). Benchmark temperature microcontroller for process dynamics and control. *Computers & Chemical Engineering, 135*, 106736.

Passino, K. M., Yurkovich, S., & Reinfrank, M. (1998). Fuzzy control (Vol. 42): Citeseer.

Rossiter, J. (2017). Using interactive tools to create an enthusiasm for control in aerospace and chemical engineers. *IFAC-PapersOnLine, 50*(1), 9120-9125.

Rossiter, J., Pope, S., Jones, B. L., & Hedengren, J. (2019). Evaluation and demonstration of take home laboratory kit. *IFAC-PapersOnLine, 52*(9), 56-61.

Simulink, C. T. f. M. a. (2021). Temperature Control of a Lightbulb. Retrieved from https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_Lightbulb

Singhala, P., Shah, D., & Patel, B. (2014). Temperature control using fuzzy logic. arXiv preprint arXiv:1402.3654.

Wei, W. (2018). On disturbance rejection for a class of nonlinear systems. Complexity, 2018.

Wu, H., Su, W., & Liu, Z. (2014). PID controllers: Design and tuning methods. Paper presented at the 9th IEEE Conference on industrial electronics and applications.