

Teaching PID Auto-Tuning Using a Low-Cost Control Kit*

DOGAN IBRAHIM

Department of Biomedical Engineering, Near East University, Faculty of Engineering, Lefkosa, Nicosia, Cyprus.
E-mail: dogan@neu.edu.tr

JAMAL F. ABU HASNA

Department of Electrical & Electronic Engineering, Near East University, Faculty of Engineering, Lefkosa, Nicosia, Cyprus

This paper describes the development of a low-cost home-made microcontroller based temperature control system. The system has been developed to support the practical sessions in undergraduate teaching of the automatic control theory course at the Near East University. The thermal plant was made from a plastic box consisting of a low-voltage heater and a semiconductor temperature sensor. The temperature inside the box was controlled using a microcontroller development kit employing a PIC18F4520 microcontroller, and providing a PWM (Pulse-Width-Modulator) output for the heater. Various digital controller algorithms and PID tuning methods can be programmed and tested using the kit. The developed kit enables students to control a laboratory scale thermal plant by learning to program the microcontroller in C. In addition, students can learn the principles and application of the important relay-based PID auto-tuning algorithm to control the plant.

Keywords: teaching control; microcontroller based control; PID auto-tuning; temperature control

1. Introduction

Laboratory sessions are a very important part of every engineering course. Students learn the theory in classes and apply their knowledge to practice by using real equipment in laboratory sessions. For example, electronic engineering students learn the complex theory of transistor amplifiers in the classroom. Then they are given assignments in laboratories where they obtain the required electronic components and construct a transistor amplifier. Students then use real physical voltmeters, multimeters, and oscilloscopes to analyse the behaviour of the constructed amplifier circuit. In order to support theoretical concepts explained in classroom, laboratory sessions must be provided to reinforce this knowledge. According to Rickel [1], students retain 25% of what they hear, 45% of what they hear and see, and 70% if they use the 'learning-by-doing' method.

Although laboratory experiments are very useful they have some problems associated with them:

- laboratory equipment can easily be damaged, for example by dropping or by misusing it;
- the characteristics of real equipment can change with age and temperature;
- laboratory equipment is costly to purchase and maintain;
- an instructor should be present in the laboratory to make sure that students setup the experiment correctly to avoid any shock from the mains electric;
- laboratory equipment needs calibration from time to time and this can be costly.

Simulation is an alternative to real experiments. In control engineering Matlab-Simulink [2, 3] is the most widely used software simulation tool. Students can create a model of the system to be simulated by using transfer function blocks, summing points, test inputs, and source and sink devices. The system model can then be simulated by applying inputs and observing the responses graphically. Controllers can easily be modelled and the system responses can be observed. The controllers can then be tuned until satisfactory responses are obtained. Although simulation can be an invaluable tool in learning about and designing controllers, it has the following disadvantages.

- Function blocks in a simulation are ideal components and they do not take into account tolerances, aging, temperature effects, and noise.
- Simulation is only useful if a model of the system is available. It is sometimes difficult or very complex to derive models of physical systems, especially nonlinear systems.
- Using real physical equipment provides more insight into the experiment. For example, while experimenting with a motor, students can see the motor spinning and can feel the effect of various changes in the system. When the same system is simulated there is no 'feel' of the motor action and students simply see a graph of the system response.

Some companies provide laboratory scale control kits that can be used to teach the practical aspects of control engineering. Feedback Instruments [4] is one of the leader companies in the design and

manufacture of control kits for students. The company offers kits such as the classical 'Ball & Beam', 'Magnetic levitation', 'Coupled tanks', 'Servo trainer', 'Digital pendulum', 'Process instrumentation' and many more. Festo Didactic [5] supplies a large range of advanced educational process control systems for learning in higher education laboratories. In addition the company offers e-learning and advanced distant courses on control engineering. Leybold [6] supply a range of laboratory teaching kits for subjects such as physics, chemistry, biology, and automatic control.

The problem with the commercially available control system kits is that they are very expensive and the cost is usually beyond the budget of many institutions. A different approach adopted by many university researchers is to develop their own control system kits, the main advantage being a lower cost. Potsaid and Wen [7] describe the mechanical design of a reconfigurable educational robotic kit that can be used to perform both linear and non-linear control system experiments. This kit is rather expensive as it requires many mechanical parts, such as motors, gyros, gearing, bearings and linkages. Ibrahim [8] describes the development of a low-cost temperature control kit based on a microcontroller. This kit can be used by developing simple controllers. Leva [9] explains details of an experimental laboratory that was set up in Milan to provide laboratory experiments in automatic control to undergraduate students. Wellstead [10] describes the teaching of automatic control theory using laboratory scale models. Here, the development and use of various PC based control engineering teaching kits are described and their links to conventional lecture-based teaching are analysed. Most of the kits described here are rather difficult to make using off the shelf parts, but the kits are available for purchase commercially. Siller-Alcala *et al.* [11] describe the development of a number of educational process control kits for teaching automatic control to undergraduate students. The kits consist of a 'heat transfer module', a 'level control system', and an 'articulated robot arm'. Although the kits were easy to construct, use of the kits require the development of a graphical user interface for the implementation of the various controller algorithms. Durfee *et al.* [12] describe the design of two educational control kits that students can take home and carry out the experiments in their own time. A 'mass-spring-damper system' and an 'analogue filter system' are given as sample kits. The disadvantage of the presented systems is that a PC running a Visual Basic application is required to interface and control the experiments. Wadoo and Jain [13] present the development of robotic based control systems experiments using LEGO construc-

tion kits and a special software called ROBOTC. Although the total cost of the kit and the software is low (around £250), the presented system has the disadvantage that it requires a special software package to be used.

Although kits developed by other researchers can be used to experiment with various controller algorithms, none of these kits were used to teach the important topic of PID auto-tuning. One common problem with most commercially available control system kits is that they are usually very expensive to purchase and maintain. In this paper the development of a low-cost home-made temperature control system is described. The developed system is called a 'kit' as students can easily obtain the required parts, and assemble and configure the complete system at home. The kit is constructed from a plastic box, a heating element, a temperature sensor chip, and a microcontroller development board. This paper shows how the kit is used in undergraduate student control engineering laboratories at the Near East University.

2. The temperature control kit

One of the aims of this project has been to develop a low-cost closed-loop digital automatic control system where students can put their theoretical knowledge into practice. Nowadays, with the availability of low-cost microprocessors, microcontrollers, and digital computers, nearly all controllers are implemented using digital techniques. Depending on the control architecture used, the set point or the feedback loop can either be analogue or digital. Figure 1 shows the most commonly used system architecture where the set point $r(t)$ is digital and is programmed into the digital controller, and the sensor is analogue and is converted into digital and fed to the digital controller. The error signal $e(t)$ is digital and is calculated by the digital computer. The output of the controller is converted into analogue using a D/A converter and this signal drives the plant.

The closed-loop control system shown in Fig. 1 has been implemented by developing a low-cost home-made thermal plant and a microcontroller development kit including a PIC18F4520 type [14, 15] powerful microcontroller. Figure 2 shows the block diagram of the developed system. The thermal plant is very simple and consists of a small round shaped plastic food container box enclosing air, 16 cm in diameter and 15 cm high. The box is made of 1.5 mm thick plastic so that it can withstand the temperature. Because the controller algorithms were to be implemented using floating point arithmetic for accuracy, it was necessary to use a microcontroller with plenty of RAM data memory, as

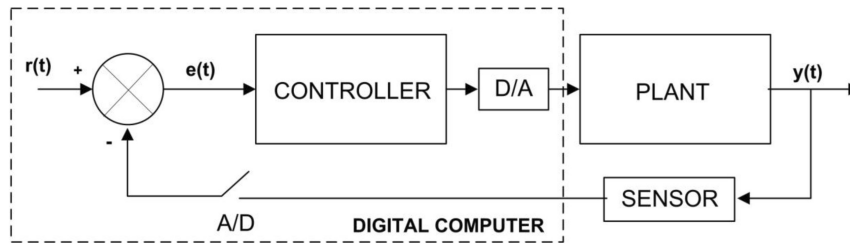


Fig. 1. Block diagram of a digital control system.

well as enough ROM program memory, and with additional on-chip features such as an A/D converter. The PIC18F4520 microcontroller used in the system satisfied all these requirements. This is a powerful microcontroller with the following basic features:

- up to 40 MHz operation
- 32K program memory
- 1536 bytes RAM memory
- 36 I/O pins
- 13 channel 10-bit A/D converters
- 8×8 multiplier hardware.

A LM35 type analogue temperature sensor chip is placed inside at the bottom of the plastic box to measure the temperature. LM35 gives an analogue output voltage directly proportional to the measured temperature, i.e. $V_o = 10 \text{ mV}/^\circ\text{C}$. The sensor voltage is converted into analogue using one channel of the on-chip A/D converter of the PIC18F4520 microcontroller. The control algorithm is implemented on the microcontroller using the mikroC language [16, 17]. mikroC is a high-level C language developed specially for the PIC microcontrollers and it provides a large number of built-in library functions, supporting various peripheral devices such as RS232, RS485, USB, CAN, I²C, SD-card, LCD, and many more. The digital control signal output from the microcontroller is converted into analogue using an external 12-bit D/A converter module connected to the microcontroller I/O ports. This signal ranges from 0 to +5 V and provides a limited current capability that cannot drive the heating element directly. Thus, a Pulse-Width-Modulator (PWM) driver module [18] is used to convert the D/A voltage to +12 V and at the same

time to provide the required high current to the heating element. The PWM module used receives an analogue input control voltage and provides a PWM output voltage whose duty cycle is directly proportional to the input control voltage. Figure 3 shows the input–output relationship of the PWM module. For example, at an input control voltage of +2.5 V the duty cycle is 25% and the average output voltage is equal to half the supply voltage i.e. about +2.5 V. The heating element was taken from an in-car portable heater used to heat the windshield in cold weather. These heaters are available for around £12.00 and are sold by most car electrical suppliers and they operate with +12 V, drawing around 6 A of current. The total electrical power of 72 W was enough to heat the plant (plastic box) to temperatures around +38°C.

The plant dynamics were measured and plotted in real time using a low-cost DrDaq [19] type data logger. This is a small data logger device that is connected to a PC via the USB port. The device has built-in sensors and dedicated inputs to measure physical quantities such as sound, light and pH. In addition, three external inputs are provided for user connected sensors. The output of the LM35 sensor was connected to one of the external inputs and the device was calibrated to directly measure the temperature. DrDaq is supplied with a software (Pico-Logger) that can be used to draw a graph of the sensor output in real time. In addition, the measured data can be logged to the disk for future analysis.

At the heart of the system there is the EasyPIC6 [20, 21] microcontroller development board. This is a sophisticated development board with on-board programmer and an on-board in-circuit debugger that can be used during program development.

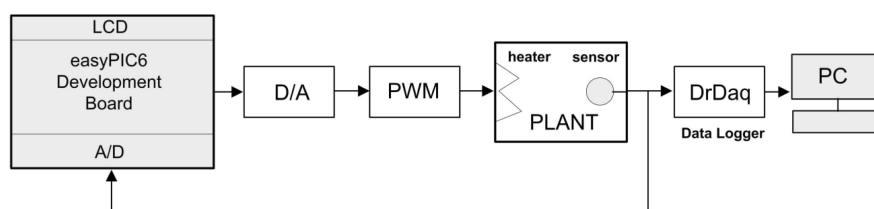


Fig. 2. Block diagram of the developed system.

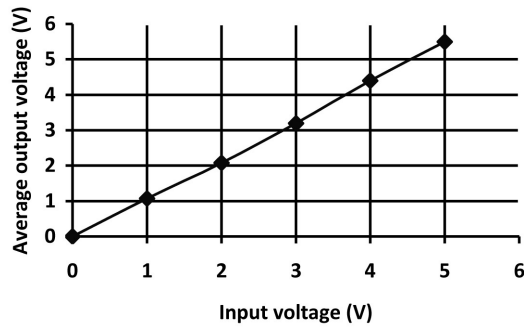


Fig. 3. Input–output relationship of the PWM.

EasyPIC6 provides an integrated development environment where a program written using the mikroC language can easily be compiled and then transferred to the development board. EasyPIC6 provides the following additional features that can be very useful during microcontroller based project development:

- sockets for 18, 20, 24, 40, 48 pin PIC microcontrollers
- 36 LEDs
- 36 push-button switches
- 2 × 16 LCD
- 128 × 64 graphics LCD
- 4 × 4 keypad and menu keypad
- RS232, PS2, and USB connectors
- all I/O pins available on connectors
- PIC microcontroller programmer
- in-circuit debugger.

The easyPIC6 board has IDC-10 type edge connectors where a large number of external modules (e.g. D/A converter) can very easily be plugged-in to the board without having to solder or make any wiring. The PIC18F4520 microcontroller is operated from an 8 MHz crystal. A 2 × 16 LCD is connected to PORT B of the microcontroller to show the state of the system in real time. The first row of the LCD shows the set point temperature, and the second row shows the actual temperature of the plant. This is very useful as the students can see the changes in the plant temperature in real time and at the same time they can compare these changes with the set point

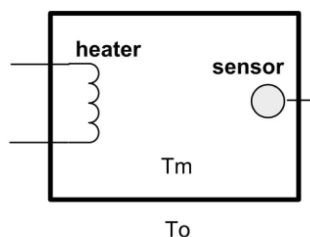


Fig. 4. Simplified model of the plant.

value. The temperature sensor is connected to the analogue input AN0 of the microcontroller. Port C pins of the microcontroller are used to drive the D/A converter, which in turn provides drive signals to the PWM module. The PWM module is powered from a +12V D.C. power supply, capable of delivering up to 10 A of current. The output of the PWM module is connected directly to the heating element.

3. The plant dynamics

The plant used in the system is a simple thermal plant with a closed plastic box and an electrical heating element used to heat the air inside the box. Before setting up the system and writing control algorithms, the students are asked to study the theoretical plant dynamics and estimate the type of behaviour expected from the system. Figure 4 shows a simplified model of the plant. Assuming that the atmospheric temperature is T_o , and the air temperature inside the box is T_m , and the heat radiated by the heater is $Q(t)$ we can write the heat transfer rate inside the box as

$$Q(t) = C \frac{dT_m}{dt} + \frac{(T_m - T_a)}{R}, \quad (1)$$

where C and R are the thermal capacitance and thermal resistance respectively. We can re-write Equation (1) as

$$Q(t) = C \frac{dT_m}{dt} + \frac{T_m}{R} - \frac{T_a}{R}. \quad (2)$$

Taking the Laplace transform and noting that T_o is constant, we can write equation (2) as

$$Q(s) = CsT_m(s) + \frac{T_m(s)}{R} - \frac{T_a}{sR}. \quad (3)$$

Figure 5 shows the block diagram representation of the plant where the atmospheric pressure acts as an external disturbance to the system.

Thermal systems always have transportation delays associated with them [22]. This is because the plant takes a certain time to respond when the heater value is changed. Adding a delay term to our system, we can represent the system as a classical first-order system with delay as shown in Equation (4). As we shall see in later sections, students are

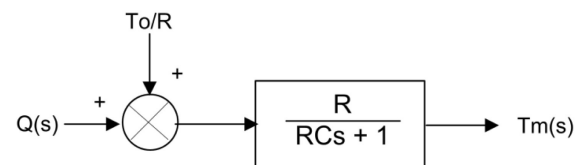


Fig. 5. Block diagram of the plant.

asked to verify the plant dynamics and find the plant parameters from experimental data.

$$\frac{T_m(s)}{Q(s)} = \frac{K e^{-T_d s}}{1 + sT}, \quad (4)$$

where T_d is the transportation delay and T is the time constant of the system.

4. Review of the PID relay auto-tuning method

The majority of controllers used in industry are of the PID type. In large processes there may be hundreds of such controllers working independently. One of the difficulties with any PID type controller is to tune the controller, i.e. to find the best controller parameters that will satisfy the requirements. The tuning procedure, if done manually, is very tedious and time consuming, especially if there are many controllers to be tuned. It is a known fact that many industrial control loops are poorly tuned, causing degradation of performance. The introduction of auto-tuning capabilities to PID controllers has shortened the time needed to commission control systems and at the same time has increased the performance and robustness of the control loop.

Before going into the details of PID auto-tuning it is worthwhile to review the basic PID controller design. The transfer function of a continuous PID controller is given by [23]

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right], \quad (5)$$

where, K_p , T_i and T_d are known as the proportional gain, integral gain, and derivative gain constants of the controller respectively. In digital control applications Equation (5) is converted into discrete form. Although there are several discrete representations of the PID, one of the most frequently used is given by the z-transform [23]:

$$\frac{U(z)}{E(z)} = K_p \left[1 + \frac{T_s}{T_i(1 - z^{-1})} + T_d \frac{(1 - z^{-1})}{T_s} \right], \quad (6)$$

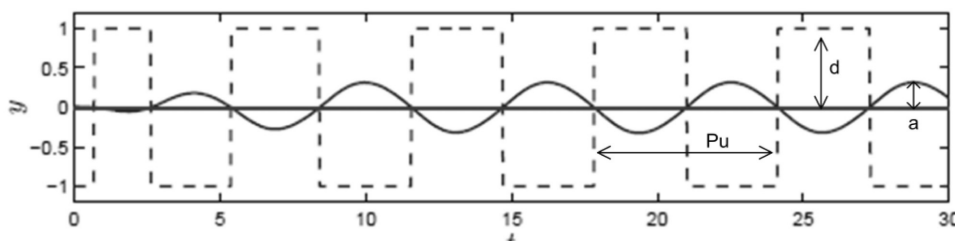


Fig. 7. Limit cycle oscillations.

where T_s is the sampling period. Implementation of Equation (6) on a digital computer is relatively easy [20] and the calculations are normally done using floating point arithmetic for higher accuracy.

Auto-tuning is the process whereby the controller parameters are found and the controller is operated with these parameters automatically on demand, usually from an operator-pressed button or an external signal. The relay feedback auto-tuning method was proposed by Astrom and Hugglund [24] and was the first commercially available auto-tuner used in industry. This method is still in use owing to its simplicity and robustness [25]. Relay auto-tuning is an efficient method for obtaining the critical point of a process and then using this information together with the classical Ziegler–Nichols rules or its variants [26] in order to find suitable PID parameters. It is important that control engineering students are familiar with the theory and application of this highly popular PID auto-tuning method.

Relay auto-tuning is based on the observation that when a relay-based nonlinearity is introduced into the forward path of a system then limit cycle oscillations will occur with a certain amplitude and period. Figure 6 shows the block diagram of the relay auto-tuner. During the tuning process the lower block is selected and during normal operation the upper block is selected. Figure 7 shows that the oscillations are approximately sinusoidal. The ultimate gain k_u (the gain that brings the system to the stability boundary) can be calculated from the limit cycle oscillations as [27]

$$k_u = \frac{4d}{\pi a}, \quad (7)$$

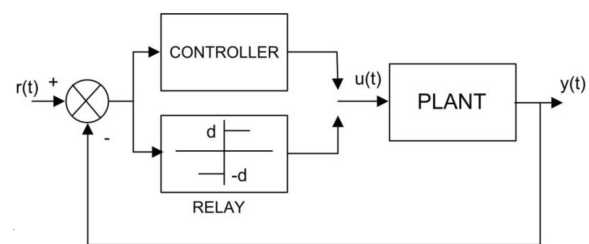


Fig. 6. Relay auto-tuner.

Table 1. Modified Ziegler–Nichols tuning constants.

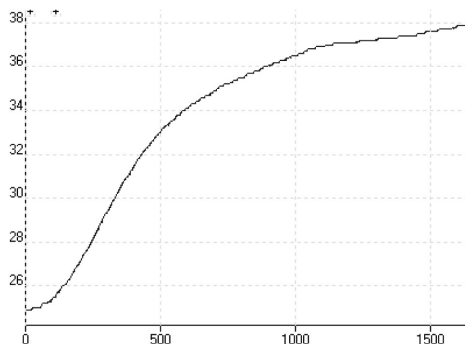
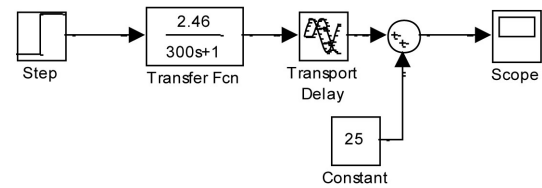
Specification	K_p	T_i	T_d
Original	$0.6K_u$	$P_u/2$	$P_u/8$
Little overshoot	$0.33K_u$	$P_u/2$	$P_u/3$
No overshoot	$0.2K_u$	$P_u/2$	$P_u/3$

where d is the relay amplitude, and a is the amplitude of the process output. Similarly, the ultimate period P_u is the period of oscillations and can easily be calculated from the response. The PID parameters can then be calculated using the modified Ziegler–Nichols rules [28] given in Table 1, or some other tuning rules. Once the PID parameters are calculated, the relay auto-tuning procedure is then completed and the controller can be put into commission.

5. The experimental study

One of the aims of this project was to teach students how to develop the relay-based PID auto-tuning algorithm by writing software on a microcontroller to control the thermal plant. Students carry out the experiment as outlined below.

- Students are asked initially to find the plant parameters using Ziegler–Nichols open-loop step response testing.
- The open-loop system response was then verified by simulation using the Matlab–Simulink program.
- The open-loop system was then converted into digital form and its response was analysed by doing various simulations.
- Students develop the relay-based PID auto-tuning algorithm to obtain the controller parameters and then they control the plant using these parameters.
- Students plot and analyse the system response by varying the controller parameters to see how the closed-loop response changes by each parameter.

**Fig. 8.** Open-loop response of the system.**Fig. 9.** Matlab–Simulink model of the open-loop system.

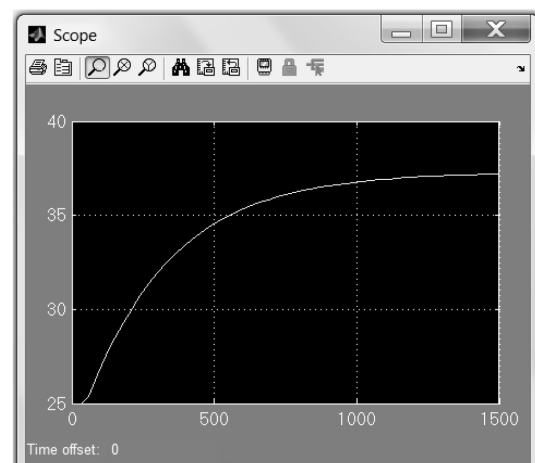
5.1 Open-loop step response and Simulink model

The open-loop step response of the system was obtained by applying a step input to the PWM module and plotting the plant temperature using the DrDaq data logger. Figure 8 shows the open-loop step response when +5 V step voltage was applied to the PWM module while the ambient temperature was 25°C. Using the Ziegler–Nichols open-loop method, the plant parameters were calculated as in Equation (8). The open-loop system response was then verified by modelling the system using the Matlab–Simulink as in Fig. 9. The simulation result, shown in Fig. 10 is very close to the actual plant response.

$$\frac{T_m(s)}{Q(s)} = \frac{2.46 e^{-50s}}{1 + 300s} \quad (8)$$

As we are using a digital computer (a microcontroller) to control the plant, it was necessary to digitize the plant and simulate the behaviour of the closed-loop response under various conditions. The Matlab function ‘c2d’ was used to digitize the plant. Using a 10 second sampling time the discrete model of the plant was obtained as shown in Fig. 11.

Students now have the ability to try various controllers and see the closed-loop plant response. This was left as an exercise for the students.

**Fig. 10.** Simulation result of the open-loop system.

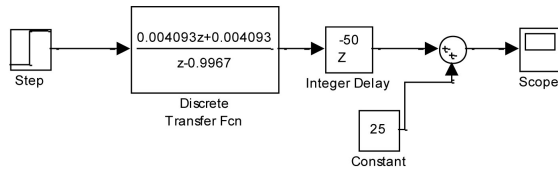


Fig. 11. Discrete model of the system.

5.2 PID auto-tuning

A nonlinear relay was introduced into the forward path and the system response was plotted. Figure 12 shows the system oscillating with a peak amplitude of 0.4°C and a period of 360 seconds. Using the values given in Table 1 the PID parameters were calculated and used to control the system. The system is put into auto-tune mode by pressing a button, and when in this mode an LED is turned ON to indicate that tuning is in progress. In this mode the amplitude and the period of oscillations are measured in real time and the PID parameters are calculated accordingly. The LED turns OFF when the tuning is complete and the system is automatically placed under control using the calculated PID parameters. The response of the system with the calculated PID parameters is shown in Fig. 13 where the set point temperature was 27.5°C . The code was written in C and Fig. 14 shows how the PID was implemented in C. An LCD display was used in the experiment to help students to see the change of temperature inside the box. The top row of the LCD showed the set point temperature and the second row showed the actual temperature inside the box in real time.

5.3 The cost

The total cost of the project consists of hardware parts and software. The hardware parts, except the EasyPIC6 microcontroller development board cost no more than £20 and they can easily be obtained. The cost of the EasyPIC6 is around £80, but this is a general purpose board and it can be used for many other electronic and control projects. The software consists of the microcontroller C compiler, and the Matlab-Simulink package. The compiler is distributed free of charge with the EasyPIC6 board. The Matlab-Simulink package, although expensive, is available at most universities engaged in teaching an engineering course.

5.4 Limitations

The developed project has the limitation that because a plastic box is used as the plant, the maximum temperature that can be controlled is around 38°C . Additionally, there is no way to cool down the plant as only the heating is controlled.

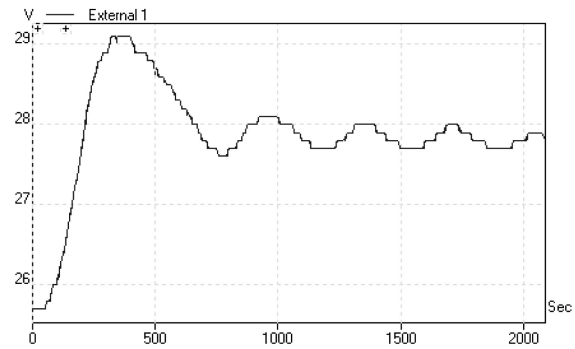


Fig. 12. System under relay auto-tuning.

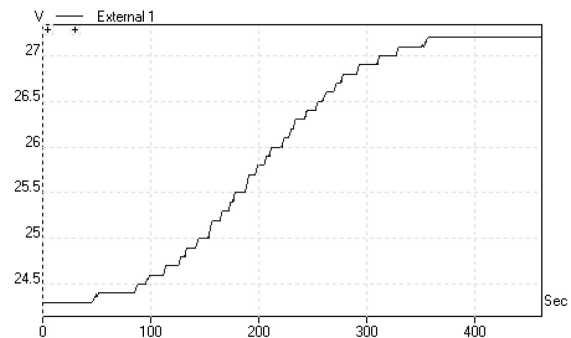


Fig. 13. Closed-loop system response with calculated parameters.

```
for(;;) // Do forever
{
    sensor = ADC_Read(0); // Get analog sensor data
    ykt = sensor*5000.0 / 1024.0; // ykt in mV
    ykt = ykt / 10.0; // ykt in Degrees C
    ekt = set_point - ykt; // Error
    pkt = b*ekt + pkt_1; // Integral term
    qkt = c*(ekt - ekt_1); // Derivative term
    ukt = pkt + a*ekt + qkt; // Control signal
    control = ukt*4095.0 / 5.0; // Convert to analog
    DAC_Output(control); // send to PWM
    pkt_1 = pkt; // Update pkt_1
    ekt_1 = ekt; // Update ekt_1
    Delay_Ms(10000); // Wait for next sample
}
```

Fig. 14. Implementing the PID algorithm in C.

6. Student feedback

The laboratory work described in this paper was carried out by all third year students (18 students) taking the course in automatic control engineering (COM341), in the spring term of the academic year 2009–2010. In order to assess the laboratory work students were given the chance to fill out a questionnaire to express their opinions. A five-point Likert scale was used with points 1, 2, 3, 4, and 5 representing ‘strongly disagree’, ‘disagree’, ‘neither agree nor disagree’, ‘agree’, and ‘strongly agree’. The following questions were asked in the questionnaire.

- I clearly understood the aims of the experiment and studied the theory of PID control algorithms before practising with the microcontroller based temperature control kit.
- The microcontroller based temperature control kit helped me to understand the practical aspects of controller design and implementation.
- The laboratory exercise was easy to perform and I had no problems during the practical session.

The survey results indicated a positive feedback to all the questions. It was found that the majority of students (92%) 'strongly agree' to the first question and they understood the aims of the experiment. The remaining 8% selected 'agreed' for this question. The results obtained from the second question are interesting as nearly all the students (97%) selected 'strongly agree'. This result emphasizes the importance of practical sessions in engineering courses. Most of the students (52%) selected 'agree' to the last question indicating that it was easy to perform the experiment and they had no difficulties during the practical session.

7. Conclusions

The design and development of a low-cost microcontroller based temperature control project has been described. The aim of the project has been to teach the practical aspects of designing controllers, and in particular PID type controllers and auto-tuning techniques to undergraduate students. Students learned how to develop PID based real-time programs using the C high-level language with microcontrollers. The developed thermal plant was constructed from a cylindrical shaped plastic box, with a heating element attached inside the top lid of the box, and a semiconductor sensor attached to the bottom of the box. A PIC18F4520 type high-end microcontroller was employed to realize the controller in which students experimented different classical PID tuning algorithms, including relay auto-tuning. Students were given the theory of the controller algorithm and they were expected to develop and implement this controller algorithm on the system using the C language.

The presented work is innovative as the project is based on using the important topic of PID auto-tuning. In addition to experimenting with conventional PID algorithms, students learn how to obtain the controller parameters using the relay-based auto-tuning algorithm.

The results were satisfactory as the temperature inside the box was controlled and reached the set point with little or no overshoot. Students learned the practical aspects of how to write code to implement the relay-based PID auto-tuning algorithm.

The easyPIC6 microcontroller development board used in the experiment has been extremely useful as it provided an integrated development environment, minimizing the requirement for any additional hardware or software.

A survey carried out at the end of the laboratory experiments indicated that the majority of students clearly understood the aims of the experiment and they were happy about the results and the way the experiment was conducted.

References

1. J. W. Rickel, Intelligent computer-aided instruction: A survey organized around system components, *IEEE Trans Syst Man Cybernet*, **19**, 1989, pp. 40–57.
2. A. Gilat, *Matlab: An Introduction with Applications*, Wiley, USA, 2010.
3. H. Klee and R. Allen, *Simulation of Dynamic Systems with Matlab and Simulink*, CRC Press, USA, 2011.
4. Feedback Instruments web-site: <http://www.feedback-group.com>, accessed 2 June 2011.
5. Festo Didactic web-site: <http://www.festo-didactic.com>, accessed 5 June 2011.
6. Leybold web-site: <http://www.ld-didactic.de>, accessed 5 June 2011.
7. B. Potsaid and J. T. Wen, Eubot: a reconfigurable kit for control education, *Proc of IEEE Int. Conf. on Control Applications*, USA, 2000, pp. 50–55.
8. D. Ibrahim, Teaching digital control using a low-cost microcontroller based temperature control kit, *Int. J. elect Eng. Educ.*, **40**, 2003, pp. 175–187.
9. A. Leva, A hands-on experimental laboratory for undergraduate courses in automatic control, *IEEE Trans. On Education*, **46**(2), 2003, pp. 263–272.
10. P. E. Wellstead, Teaching control with laboratory scale models, *IEEE Trans. on Education*, **33**(3), 1990, pp. 285–290.
11. I. I. Siller-Alcala, J. Jaimes-Ponce, R. Alcantara-Ramirez and J. J. Rubioavila, a hands-on process control laboratory for undergraduate students: using low cost experiment kits, *Proc of the 7th WSEAS International Conference on Education and Educational Technology (EDU'08)*, 2008.
12. W. Durfee, P. Li and D. Waletzko, Take-home lab kits for system dynamics and controls courses, *Proc of the 2004 American Control Conference, Vol. 2*, 2004, pp. 1319–1322.
13. S. A. Wadoo and R. Pain, A LEGO based undergraduate control systems laboratory, *IEEE Systems, Applications, and Technology Conference (LISAT, 2012)*, 2012, pp. 1–6.
14. PIC18F4520 Data sheet, web-site: <http://www.microchip.com>, accessed 5 June 2011.
15. B. B. Brey, *Applying PIC18 Microcontrollers: Architecture, Programming, and Interface Using C and Assembly*, Prentice-Hall, UK, 2007.
16. mikroC User Guide, web site: <http://mikroe.com>, accessed 2 November 2010.
17. D. Ibrahim, *Microcontrollers for Students—The Theory*, Anchor Print Group, UK, 2011.
18. Pulse width modulator K8004 User manual, web-site: <http://www.apogeekits.com>, accessed 20 May 2011.
19. USB DrDAQ data logger manual, web-site: <http://www.drdaq.com>, accessed 15 May 2011.
20. D. Ibrahim, *Microcontrollers for Students—Practicals*, Anchor Print Group, UK, 2011.
21. EasyPIC6 User Guide, web-site: <http://www.mikroe.com>, accessed 2 May 2010.
22. N. S. Nise, *Control Systems Engineering*, Wiley, USA, 2010.
23. D. Ibrahim, *Microcontroller Based Applied Digital Control*, Wiley, 2006.
24. K. J. Astrom and T. Hagglund, Automatic tuning of PID controllers, *Instrument Society of America*, USA, 1988.

25. C. C. Hang and K. K. Sin, A Comparative Performance Study of PID Auto-tuners, *IEEE Control Systems*, **11**(5), 1991, pp. 41–47.
26. M. Zhuang and D. P. Atherton, Automatic tuning of optimum PID controllers, *IEE Proceedings*, **140**(3), 1993.
27. C. C. Hang, K. J. Astrom and Q. G. Wang, Relay feedback auto-tuning of process controllers—a tutorial review, *Journal of process Control*, **12**, 2002, pp. 143–162.
28. D. I. Wilson, Relay-based PID Tuning, *Automation & Control*, Feb/March, 2005, pp. 10–12.

Dogan Ibrahim received his B.Sc. degree in Electronics (First Class Honours) from Salford University, UK, his M.Sc. in automatic control from Manchester University and his Ph.D. in the field of digital signal processing at City University, London. His interest are in the field of automatic control, microcontroller based design, and digital signal processing.

Jamal Fathi Abu Hasna obtained his BSc degree in Electrical and Electronics Engineering from Near East University in 1998–99. He completed his MSc degree in 2001, and then continued to his Doctoral studies at the same university. His current interests include cellular communications, neural computing, adaptive signal processing, control systems, and wavelets.