

# Computer based learning in an undergraduate physics laboratory: interfacing and instrument control using Matlab

**J S Sharp, P M Glover and W Moseley**

School of Physics and Astronomy, University of Nottingham, University Park, Nottingham, NG7 2RD, UK

E-mail: [james.sharp@nottingham.ac.uk](mailto:james.sharp@nottingham.ac.uk) and [paul.glover@nottingham.ac.uk](mailto:paul.glover@nottingham.ac.uk)

Received 11 August 2006, in final form 6 November 2006

Published 30 April 2007

Online at [stacks.iop.org/EJP/28/S1](http://stacks.iop.org/EJP/28/S1)

## Abstract

In this paper we describe the recent changes to the curriculum of the second year practical laboratory course in the School of Physics and Astronomy at the University of Nottingham. In particular, we describe how Matlab has been implemented as a teaching tool and discuss both its pedagogical advantages and disadvantages in teaching undergraduate students about computer interfacing and instrument control techniques. We also discuss the motivation for converting the interfacing language that is used in the laboratory from LabView to Matlab. We describe an example of a typical experiment the students are required to complete and we conclude by briefly assessing how the recent curriculum changes have affected both student performance and compliance.

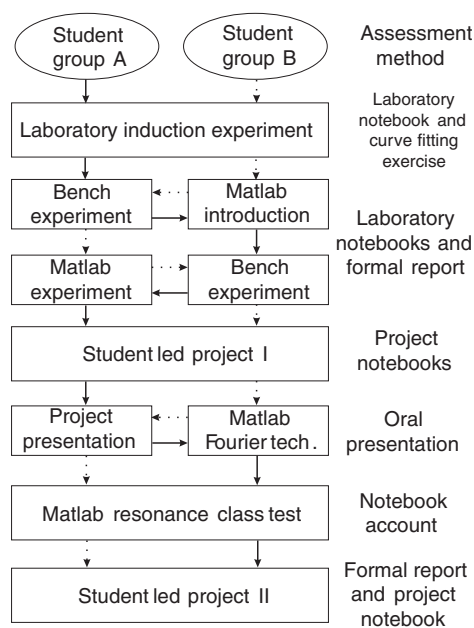
## Introduction

Computers are becoming an increasingly common part of everyday life and producing computer literate graduates is a key concern for any modern university physics department. Recent technological advances have meant that computers are increasingly being used by scientists to collect data and automate experiments, particularly as the necessary equipment is becoming cheaper and more accessible. Understanding the principles of computer interfacing techniques therefore provides science graduates with a competitive advantage in the workplace. A number of computer based interfacing and process control based laboratories have been recently developed to address the increasing need for computer literate experimental scientists and engineers [1–6]. However, experience has shown that students do not always appreciate the value of learning interfacing techniques and can often see them as being less valuable than more ‘hands-on’ or ‘pure physics’ experiments.

The advantages of using computers to perform measurements and automate data collection are obvious. The ability to monitor rapid processes with sampling rates of the order of 100 kHz means that students can routinely acquire and analyse data from rapidly changing physical systems (which would otherwise be impossible) using commercially available data acquisition (DAQ) cards. In addition to this, instrument functions such as lock-in detection and RMS calculations can be performed without the need for additional and expensive hardware. These techniques also provide a large pedagogical advantage over more traditional 'hands-on' based methods. The ability to change a parameter in a program and determine the resulting effect on the physical system of interest is a powerful tool in helping students to develop a deeper understanding of the physical principles that are being presented [7]. However, great care also has to be taken to ensure that students avoid the pitfalls associated with trusting every result that is generated by a computer [8]. To overcome these difficulties we have found it necessary to teach these powerful interfacing techniques both in combination with (and in the context of) more traditional physics experiments. This ensures that students develop the necessary practical skills and the appreciation of experimental errors and uncertainties that they will need to assess the results of their own experiments in a clear and critical manner.

The second year undergraduate physics laboratory course at Nottingham is made up of two components. The first teaches the students how to perform traditional 'hands-on' bench-top experiments and teaches them the necessary skills that are required to enable them to treat experimental uncertainties and to reflect critically upon their results. The second component teaches the students how to use interfacing methods to control experiments and to acquire and analyse data using a PC. Until recently, the two parts of the course were taught separately and it was found that students were becoming increasingly frustrated with the course. In particular, the interfacing component had become increasingly unpopular with students. This was due to a number of factors relating to the way in which the course was delivered, the student's unfamiliarity with the programming environment being used (LabView, National Instruments) and varying levels of computer literacy in each student cohort. It was found that students were spending too much time dealing with the intricacies of computer programming in LabView and were spending less time understanding and developing the key physical concepts being presented. They were also struggling to recognize the relative importance and complimentary nature of interfacing techniques and traditional physics experimental methods. As a result of this feedback, the decision was taken to teach the two parts of the course in parallel. The changes in the course structure were also combined with the introduction of a text based programming language (Matlab, MathWorks) as the main interfacing language. Students had already learned to program in Matlab in the first year of their physics course and the use of a familiar language was intended to alleviate some of the problems associated with learning an often totally unfamiliar programming language. It was also hoped that these steps would help students to reinforce the importance of using both traditional 'hands-on' experiments and computer based interfacing methods in physics experiments, seeing them as integrated rather than separate aspects of experimentation.

In this paper we give an overview of the course structure and describe a recently introduced interfacing experiment based around a thermoelectric (Peltier) device. This experiment demonstrates acquisition and control coupled with aspects of both semiconductor and thermal physics. We then discuss the relative advantages and disadvantages of using Matlab as an educational tool and compare it directly with its predecessor, LabView. We also briefly describe how student's perceptions of the laboratory have changed since the integration of the course and the introduction of Matlab as the interfacing language.



**Figure 1.** Schematic diagram showing how students progress through the second year undergraduate physics laboratory course at Nottingham. The solid and dotted lines show the progress of groups A and B respectively. Details of the methods that are used to assess students progress at each stage of the course are given in the right-hand column.

## Course overview

The structure of the student's second year is one of a movement from prescriptive training through to independent project style work (i.e. taking them from the end of their first year to the start of their third; see figure 1). The year's laboratory work started with a lecture about safety and conduct within the laboratory, the expected structure of laboratory diaries and formal reports and a reminder of how students should deal with errors and uncertainties. The laboratory induction session involved the use of a familiar experiment that was based on a bar pendulum. This experiment was designed to refine the student's abilities in dealing with experimental uncertainties and helped to build upon concepts relating to standard error and compound uncertainties that the students developed in their first year. Prior to starting the second year laboratory course, students had been instructed in the use of maximum possible error (MPE) approaches when combining experimental uncertainties. At the beginning of the second year course, we insisted on the use of the standard error approach with weighted uncertainties being added in quadrature. We specifically target these aspects during the induction laboratory class and cover the importance of planning experiments, writing diaries and making sensible estimates of uncertainties. This introductory session was also used to reinforce the methods that we expect students to use when dealing with the propagation of uncertainties using analytical functions and graphical methods. The students also carried out a worksheet-based exercise to reinforce ideas relating to normal distributions, the treatment of uncertainties and to introduce them to nonlinear fitting (including the relevance of fit parameter confidence intervals).

The students were then split into two groups. These groups took a slightly different route through the course, but covered the course material at approximately the same time.

This avoided problems relating to the availability of laboratory and interfacing equipment. A summary of the different routes taken by each of the groups is given in figure 1 along with the assessment methods that were used at each stage in the course. The first semester's work comprised a series of individually performed 'hands-on' bench-top experiments and Matlab based interfacing exercises that were assessed using the students laboratory notebook accounts of the experiments and a formal written report. The bench experiment component of the module included experiments using laser interferometry, measurements of dielectric constants and dipole moments in simple liquids and measurements of black-body radiation and the determination of the Stefan–Boltzmann constant. The Matlab component of the course began with a single laboratory session on an introduction to basic programming concepts in Matlab, as well as some of the more specialized functions such as curve fitting, plotting data and saving data to file. This was then followed by one week of learning the basics of digital input/output techniques as well as an in-depth description of how to use the analogue input and output functions in Matlab. The students were then asked to use these skills to perform experiments where they provide control signals and use simple triggering methods to allow them to collect experimental data.

The second semester was based on two student-led investigations (that were carried out in pairs), where only basic information was given on theory and equipment. These project style investigations were based on either Matlab interfacing experiments or more traditional experiments where students took readings and recorded them in their laboratory notebooks. Each student was required to complete a project of each type. Having been assigned their projects, the students were required to plan and carry out the experimental investigations themselves using the skills that they had developed in the first semester. In addition to the investigations, the students also received some further training in the use of Fourier transforms and signal processing methods. These were timetabled in such a way that they complemented the Fourier applications synoptic module that the students take in the second semester. All of the students' interfacing and signal processing knowledge was then tested using a resonance-based class test. This test was performed under examination conditions and lasted for 3 h. The assessment of this element was based on the students' laboratory notebook account of the experiment, their Matlab code and the experimental data collected. Details of the majority of the interfacing experiments (as they were previously taught using LabView) can be found in a publication by Moriarty and co-workers [1]. These included an experiment that uses Matlab to determine the thermal conductivity of copper. In this experiment, triggered data acquisition was used to measure the time dependence of changes in temperature of the surfaces of copper disks that were heated on one side using a flash gun. In addition, students are asked to perform *experimental investigations* where Matlab was used to help study phenomena such as optical absorption in III–V semiconductors, the photoelectric effect, ferromagnetic hysteresis, 'thermal noise and the determination of Boltzmann's constant' and chaotic behaviour in a bouncing ball electrical circuit.

In the following section we describe one of the new Matlab based interfacing experiments that is used on the current version of the course. This experiment allows students to study the thermoelectric properties of a Peltier device and enables them to develop an understanding of some of the basic concepts related to control theory and the use of the Peltier device as a temperature controller.

### **Thermoelectric effect and the Peltier device**

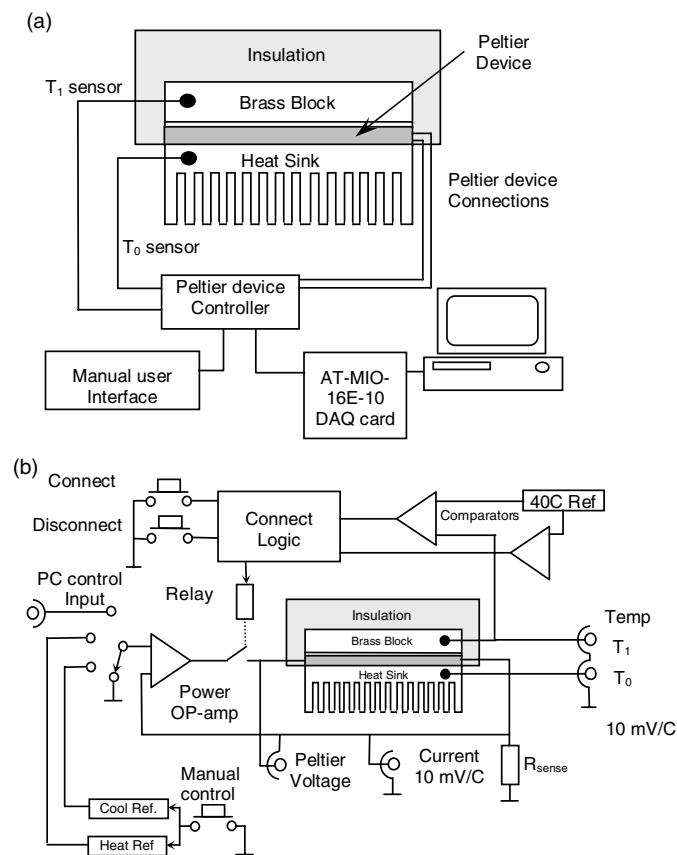
The Peltier device uses the thermoelectric effect to maintain a temperature difference between two of its surfaces. These devices are manufactured by connecting a number of semiconducting

(alternating p- and n-type) elements in series and in such a way as to give a large effective surface area at the junctions between the two types of material. The device generates an output voltage that is proportional to the difference in temperature between its surfaces and an electrical current will flow if the device is placed in a circuit. Conversely, if a current is driven through the device this will result in the generation of a temperature difference between the junctions. The principle that governs the operation of these devices is the exactly the same as that which underlies the operation of thermocouples.

In this experiment, students were asked to study the thermoelectric effect in a commercially available Peltier device. The students were required to develop an understanding of the physics of the operation of these devices and then to use the relevant equations to determine how parameters such as the Seebeck coefficient and thermal conductivity of the device could be determined (see below). They were then asked to use Matlab to write software that would enable them to monitor the temperature of the two surfaces of the device, as well as the device current and voltage in real time. Following this, the students were asked to plot out the data and to use the built in Matlab fitting routines to determine the relevant material parameters for the device along with their associated uncertainties. Having done this, the more talented students were then asked to use Matlab to write a software based temperature control program to explore the properties of different types of temperature controllers. This involved the use of *on/off*, *proportional* and *integral* control based methods to maintain a required temperature difference between the two surfaces of the device. These exercises allowed students to build upon the knowledge of Matlab that they acquired in the introductory Matlab sessions and to apply it to a real physical problem.

#### *Experimental apparatus*

A schematic diagram of the equipment used in the Peltier experiment is shown in figure 2(a). A 30 mm × 30 mm × 9 mm brass block was attached to the top surface of the Peltier device (CP1.0-127-05 L Melcor Corporation, Trenton, NJ, USA) [9]. A commercially available heat-sink compound was used to ensure a good thermal contact. The Peltier device was then mounted on a 1.9 °C W<sup>-1</sup> heat sink (Aavid Thermalloy Concord, NH, USA) to ensure that the bottom surface of the device was maintained at room temperature. Temperature sensors (LM35DZ, National Semiconductor, Santa Clara, CA, USA) were mounted in holes drilled in the brass block and the heat sink to measure the temperatures of the top and bottom surfaces of the Peltier device,  $T_1$  and  $T_0$  respectively (see figure 2(a)). The brass block and Peltier device were then enclosed in polystyrene foam insulation and covered by a plastic box to hold the assembly in place. The temperature sensors were then connected to the Peltier device control box along with the connections that measured the Peltier device current  $I$  and voltage  $V$ . The control box provided the user interface, device protection and manual control functions needed to perform the experiments (see figure 2(b)). The power operational amplifier shown schematically in figure 2(b) was based on an OP27 integrated circuit with a BD675/6 Darlington complimentary pair arranged as a voltage follower. Current feedback was provided by a 0.1Ω current sense resistor and the input to the amplifier could be switched between four different settings to provide different heating characteristics. The *Zero* setting gave a zero current through the Peltier device, *Cool* and *Heat* settings (enabled by a manual push button) generated a 0.2 A current in the device with an appropriate polarity that could be used to lower or raise the block temperature  $T_1$ . Direct control of the device voltage/current could also be provided using a low current power supply or one of the analogue outputs of a DAQ card installed in a PC by using the *PC* setting on the Peltier control box. The device voltage, current and device temperatures were all measured using connections to insulated



**Figure 2.** Schematic diagram of the experimental setup used in the Matlab based Peltier experiment. The Peltier device is connected to a control box that has both manual and PC control options (panel (a)). Panel (b) shows a schematic wiring diagram of the Peltier device controller used in these experiments.

BNC sockets on top of the Peltier control box. These were then connected to differential inputs of a AT-MIO-16-E-10 data acquisition (DAQ) card (National Instruments) installed in a PC (see figure 2(a)). The temperature sensor voltages were calibrated at  $10 \text{ mV } ^\circ\text{C}^{-1}$  and were compared to a  $400 \text{ mV}$  ( $40 \text{ }^\circ\text{C}$ ) reference. When this reference temperature was exceeded the control logic circuitry de-activated a relay that disconnected the power amplifier from the Peltier device. This was done to prevent the device overheating. The power amplifier could also be disconnected manually using a button on the Peltier control box. This method of disconnecting the power amplifier was found to be valuable as it ensured that the device response could be monitored under zero current conditions during an experiment (see below).

#### *Theory of operation*

The Peltier device relies on the principle that a piece of semiconductor will develop a potential difference across it when a temperature gradient is applied to the device. A description of the

theory underlying the thermoelectric effect and the operation of Peltier devices can be found elsewhere [10–12]. The main results are briefly summarized below.

The electric field,  $E$ , that develops across an open circuit piece of semiconductor when a temperature difference  $\Delta T$  is applied across it is given by  $E = \alpha \Delta T$ , where  $\alpha$  is the Seebeck coefficient of the material. An interesting consequence of this result is that an applied field will cause a current to flow through the semiconductor and the movement of charge gives rise to a temperature gradient whose sign depends upon the nature of the majority charge carriers in the material (e.g. electrons or holes for n- and p-type materials, respectively) [10]. The Peltier device comprises slabs of n-type and p-type semiconductor material that are connected in series using thin metal contacts. As a result of this, the junctions between the semiconductors do not form a barrier potential as they would do in a diode and electrical conduction can occur freely in both directions. In a Peltier device, the elements are arranged in such a way that the heat transfer caused by a flowing current occurs in the same direction. This results in one side of the device being heated/cooled with respect to the other, when a current is passed through it.

The amount of heat that is transported from the ‘hot’ to the ‘cold’ junctions via thermal conduction in the device,  $P_T$ , is proportional to its thermal conductivity  $\kappa$  and the temperature difference between its surfaces. As mentioned above, the electrical power dissipated in the metal contacts between the two semiconductor materials,  $P_e$  flows equally into both the p- and n-type materials. The thermoelectric effect also results in the transfer of heat between the two surfaces of the device and the Peltier coefficient ( $\Pi$ ) is used to define the heat energy carried per unit charge in a material. This is in turn related to the Seebeck coefficient of the material by the expression  $\Pi = \alpha \Delta T$ . The corresponding thermoelectric heating power is then given by  $P_\Pi = \Pi I$ , so that the total heat pumped away from the cooled junction (referred to as  $T_1$ ) is given by

$$P_1 = P_\Pi - \frac{1}{2}P_e - P_T = 2N(\alpha I(T_1 - T_0) - I^2\rho/2G - \kappa(T_1 - T_0)G) \quad (1)$$

where  $\rho$  is the resistivity of the material between the p- and n-type semiconductors,  $N$  is the number of series elements and  $G$  is a geometrical constant of the device (measured in metres) [9]. The voltage across the device,  $V$ , is given by summing over the voltages across all elements and gives

$$V = 2N(I\rho/G - \alpha(T_1 - T_0)) \quad (2)$$

where the first term represents the ohmic response of the metal contacts and the second term describes the thermoelectric contribution of the junctions in the device.

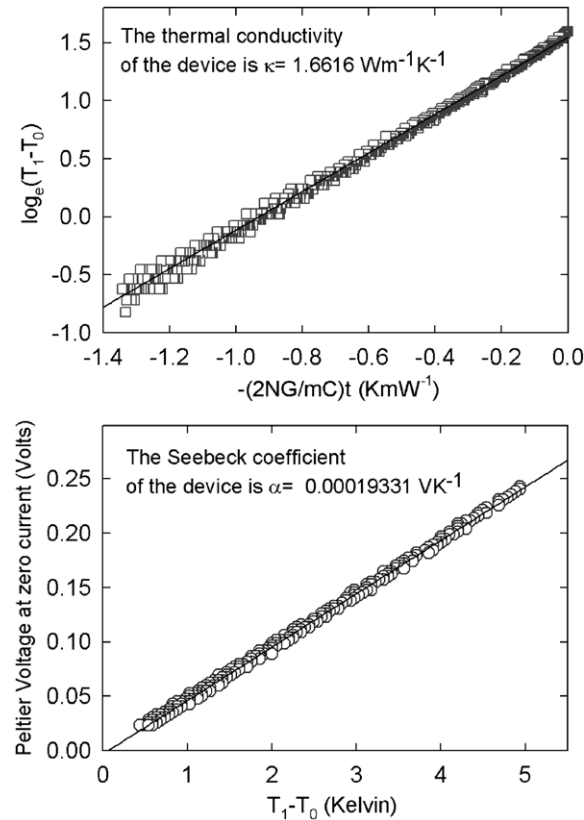
Assuming zero current conditions and that all the heat energy that is ‘pumped’ away from the cold junction results in a change in the temperature of the brass block attached to the top surface of the device (see figure 2(a)), equations (1) and (2) reduce to the following forms:

$$\frac{d(\Delta T)}{dt} = -\frac{2N\kappa G}{mC}\Delta T \quad (3)$$

$$V = -2N\alpha\Delta T \quad (4)$$

where  $\Delta T = (T_1 - T_0)$  and  $m$  and  $C$  are the mass and specific heat capacity of the brass block, respectively. The solution to equation (3) has the form of an exponential decay with a time constant that is proportional to  $1/\kappa$  and equation (4) shows that a plot of the device voltage as a function of the temperature  $\Delta T$  produces a straight line with slope  $\alpha$  (see figure 3).





**Figure 3.** Examples of data collected during the time dependent decay of the temperature difference between the two surfaces of the Peltier device under zero current conditions (i.e. device disconnected). Initially, the manual heat button was used to increase the temperature difference between the two device surfaces. The device was then disconnected (see figure 2(b)) and the temperatures ( $T_1$  and  $T_0$ ), device voltage ( $V$ ) and current ( $I$ ) were monitored as a function of time. Panel (a) shows a semi-logarithmic plot of the difference in temperature between the two surfaces of the device ( $T_1 - T_0$ ) as a function of the reduced time  $-\frac{2NG}{mC}t$ . Panel (b) shows data obtained for the variation of the modulus of the Peltier device voltage as a function of the temperature difference between the two surfaces of the device.

## Discussion

In this experiment students were asked to determine how they could use measurements obtained from the Peltier device to determine the Seebeck coefficient and thermal conductivity along with their associated uncertainties. The students were told to assume that the heat pumped from one side of the device appeared on the other (i.e. there are no losses from the edges of the device) and that the system was thermally isolated from the surrounding environment. They were also given hints about monitoring the response of the device under zero current conditions and were asked to write suitable Matlab programs that would allow them to automate the data collection and analysis.

A consideration of equations (3) and (4) shows that it is possible to determine the Seebeck coefficient and thermal conductivity of the device by monitoring the temporal response of the device voltage and the temperature difference between the two device surfaces under zero current conditions (i.e. with the device disconnected).

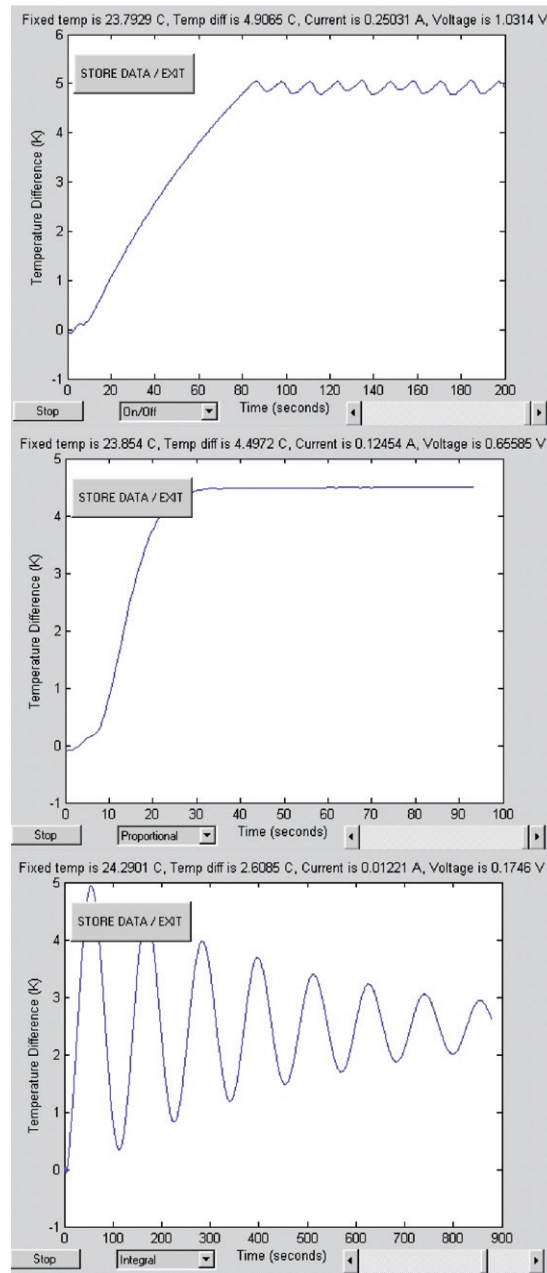


A typical example of data that were collected from the Peltier device under these conditions are shown in figure 3. The bottom surface of the Peltier device was maintained at a temperature close to room temperature ( $T_0 \sim 25^\circ\text{C}$ ) and the initial temperature of the top surface of the device (prior to cooling) was typically  $T_1 \sim 30^\circ\text{C}$ . Typical values of  $\alpha$  and  $\kappa$  that were determined from the experimental data measured during the cooling of the Peltier device were  $(1.93 \pm 0.02) \times 10^{-4} \text{ V K}^{-1}$  and  $(1.66 \pm 0.03) \text{ W m}^{-1} \text{ K}^{-1}$ , respectively. These compare favourably with generic device values of  $\alpha = 2 \times 10^{-4} \text{ V K}^{-1}$  and  $\kappa = 1.5 \text{ W m}^{-1} \text{ K}^{-1}$  taken from the data sheet [9].

The more capable students often completed this initial exercise well before the end of the 6 h laboratory session and were asked to use Matlab to write a software based temperature controller that used different methods of control to maintain a temperature difference between the two device surfaces. This was done by using an appropriate control loop structure in Matlab and by applying a control signal to the PC input of the device control box using one of the analogue outputs from the AT-MIO-16E-10 DAQ card installed in the PC. Typical examples of data that were collected using three different control methods (*on/off*, *proportional* and *integral*) are shown in figure 4. Simple adjustments of the parameters used within the control software (such as the error multiplier) have an easily recognizable effect on the transient response, steady-state error and ability of the controller to maintain a stable temperature difference between the surfaces of the device. Although the experiment could easily be adapted to demonstrate aspects of control theory, such as the Nyquist stability criterion, we do not pursue these concepts any further in the context of this laboratory [13].

The two exercises described above proved to be extremely valuable in helping students to apply the basic interfacing techniques that they learn in the introductory Matlab sessions. These exercises allow the students to learn how to combine the data acquisition capabilities of Matlab with its powerful data manipulation, curve fitting and analysis routines and enable them to discover the power of using computer-based data acquisition and analysis in physics experiments. The students that were able to use Matlab to write a software-based temperature controller also learned the basic principles of control theory and developed an understanding of the advantages and disadvantages of using different types of process control methods.

There are a number of advantages to using Matlab to perform these experiments instead of its predecessor LabView. Firstly, the matrix manipulation routines in Matlab are easy to use and overcome the difficulties that were associated with the use of different data types in LabView [1]. This means that students were able to grasp the key aspects of data analysis very quickly as only a basic knowledge of matrix algebra was required. Similarly, the plotting commands in Matlab are relatively simple and intuitive to use. Moreover, they allow students to include error bars on plots of experimental data. The inability to include error bars on plots in the version of LabView that was used in the laboratory was seen a major disadvantage when trying to encourage students to consider the uncertainties associated with experimental data. The use of Matlab has also helped to avoid some of the problems relating to the student's unfamiliarity with the programming language being used. However, some of the less computer literate students can still find it difficult to implement the knowledge of Matlab that they gain from their first year programming course and often find it difficult to incorporate the new interfacing commands in to Matlab programs to allow them to automate data collection. Students that find programming in a text based language difficult tend to struggle with basic programming concepts such as loop structures and this can act as a barrier to learning the important physics concepts that are being presented. This has recently been alleviated by giving students access to a set of low level analogue input/output Matlab programs that can be modified for a particular experiment. Students are also given some guidance on which of these sample programs they should use in each experiment. These are based upon low



**Figure 4.** Plots of the time dependence of the temperature difference between the two surfaces of a Peltier device. Data are shown for three experiments where different methods of temperature control were used. In each case the control method was selected using a drop down menu in the Matlab GUI interface and the desired temperature difference between the device surfaces was set using the slider control (bottom right-hand corner). The DAQ card installed in the PC was then used to simultaneously monitor the time dependence of the device current and voltage as well as the temperature difference  $T_1 - T_0$  between the two surfaces of the device. These values were then displayed in the title line of each of the plots. Panel (a) shows data that were collected using a simple *on/off* controller. Panel (b) shows data that were obtained using *Proportional* control and panel (c) shows the temperature variation observed using an *integral* control method.

level programs that we encourage students to write in the introductory Matlab interfacing sessions. This avoids having a series of 'black box' style programs that the students just use and do not understand and encourages them to learn the basic principles of interfacing and instrument control, before they apply them to an interfacing experiment. These programs allow the less capable students to participate in the interfacing experiments and to develop an understanding of the physics being presented without being repeatedly penalized for not being able to program.

Comparison of the student feedback that we have received from the Matlab based interfacing part of course and the previous LabView based component suggests that students still find the interfacing sections of the second year laboratory difficult. One gratifying aspect of the feedback that we have received is that more students seem to have developed an appreciation of the value of using both interfacing methods and more traditional 'hands-on' based methods in experimental science as a result of completing the new course. There has also been some indication that the marks obtained by students taking the course have increased slightly over the past two years. However, comparison of the new course with the previous LabView based course is unfair, as a number of unfortunate computer hardware problems (that are not related to Matlab or the DAQ cards) have been a source of major frustration among students taking the new course. Making a comparison between the two versions of the course at this stage is also unrealistic because the previous LabView version of the course had been running for 6 years and significant problems had been addressed during this time. We are still in the process of refining and optimizing the Matlab based interfacing course and this may be reflected in the feedback that we have received so far.

Many students have used the interfacing techniques that they have learned in this laboratory for subsequent project work in their third or fourth year of study. The informal feedback obtained from discussions with these students indicates that they had acquired the skills required to enable them to get complex experiments working quickly. Fourth year MSci (Masters) project students that were taught LabView in the previous version of the course have also expressed a preference for using Matlab for data acquisition rather than LabView. However, this feedback is informal and has been collected from a relatively small group of our more capable students. As all the students that are currently taking the fourth year were taught Matlab during their second year, this opportunity is unlikely to arise again. Despite this, we find this feedback to be encouraging as it seems to indicate that the text based format used by Matlab appeals to our students more than the often unfamiliar graphical interface of LabView.

## Conclusions

Matlab is a powerful computer programming language that is a valuable tool for teaching interfacing techniques and process control methods to undergraduate physicists. The use of analogue and digital input/output commands combined with Matlab's data manipulation and analysis algorithms make it ideal for integration into an undergraduate teaching laboratory so that interfacing methods can be taught alongside more traditional 'hands-on' based experiments.

## Acknowledgments

We are grateful to Philip Moriarty and all the demonstrating staff for their comments and input during the development of the laboratory module. We would like to thank Paul Reynolds and all the workshop staff that have contributed to the design and building of the experimental

apparatus. We would also like to thank Philip Hawker for maintaining the computing facilities in the teaching laboratory.

## References

- [1] Moriarty P J, Gallagher B L, Mellor C J and Baines R R 2003 *Am. J. Phys.* **71** 1062–74
- [2] Doyle F J III, Gatzke E P and Parkers R S 1998 *Comput. Appl. Eng. Educ.* **6** 181–91
- [3] Foley E L, Candela D, Martini K M and Tuominen M T 1999 *Am. J. Phys.* **67** 389–93
- [4] Koku A B and Kaynaks O 2001 *IEEE Trans. Educ.* **44** 24–8
- [5] Kocijancic S 2002 *IEEE Trans. Educ.* **45** 26–32
- [6] Dixon W E, Dawson D M, Costic B T and de Queiroz M S 2002 *IEEE Trans. Educ.* **45** 218–26
- [7] Trumper R 2003 *Sci. Educ.* **12** 645–70
- [8] Steinberg R N 2000 *Phys. Educ. Res. Suppl. Am. J. Phys.* **68** S37–41
- [9] The data sheet for the CP series Peltier device used in these experiments can be downloaded from the Melcor Corporation website <http://www.melcor.com/cpseries.html>
- [10] Kittel Y 1996 *Introduction to Solid State Physics* 7th edn (New York: Wiley)
- [11] Glover P M and Sharp J S 2006 *2nd Year Laboratory Manual*, School of Physics and Astronomy, University of Nottingham
- [12] Godfrey Y *An Introduction to Thermoelectric Coolers Melcor Corporation*, [http://www.melcor.com/tec\\_intro.html](http://www.melcor.com/tec_intro.html)
- [13] D’Azzo J J and Houpis C H 1995 *Linear Control System Analysis and Design: Conventional and Modern* 4th edn (New York: McGraw-Hill)