AAPT PHYSICS EDUCATION AMERICAN JOURNAL of PHYSICS

Graphical computing in the undergraduate laboratory: Teaching and interfacing with LabVIEW

P. J. Moriarty, B. L. Gallagher, C. J. Mellor, and R. R. Baines

Citation: American Journal of Physics **71**, 1062 (2003); doi: 10.1119/1.1582189
View online: http://dx.doi.org/10.1119/1.1582189
View Table of Contents: http://scitation.aip.org/content/aapt/journal/ajp/71/10?ver=pdfcov
Published by the American Association of Physics Teachers

Graphical computing in the undergraduate laboratory: Teaching and interfacing with LabVIEW

P. J. Moriarty, B. L. Gallagher, C. J. Mellor, and R. R. Baines
School of Physics and Astronomy, University of Nottingham, Nottingham NG7 2RD, United Kingdom

(Received 7 October 2002; accepted 18 April 2003)

We describe the development and implementation of an undergraduate physics laboratory course based on National Instruments' LabVIEW application. LabVIEW, a graphical programming language, provides an intuitive interface with which to teach fundamental computer-based data acquisition techniques. To convey the importance of these techniques in modern experimental physics, during our course the students complete a variety of tasks and experiments based around LabVIEW virtual instruments that they have constructed. Furthermore, LabVIEW is a powerful signal processing and waveform analysis tool, it may be used to reinforce core physics concepts taught in an analytical fashion in other courses. Foremost among these is Fourier analysis. We discuss the efficacy of LabVIEW as a pedagogical tool in a number of Fourier-related areas. Other important pure and applied physics topics covered in our LabVIEW course and briefly described here include resonance, filtering and lock-in techniques, thermal diffusivity, chaos, and optical absorption in solids.
[DOI: 10.1119/1.1582189]

I. INTRODUCTION

Given the ubiquity of computer-based data acquisition and analysis in modern experimental physics, it is essential that undergraduate students gain wide experience of these methods in their laboratory courses. While one approach is simply for the students to use prewritten control programs (effectively ''black box'' software) to acquire the experimental data, this approach neglects the requirement that ultimately they should have the ability to design computer-driven experiments, taking into consideration key elements such as the resolution of analog-to-digital converters, sampling frequency requirements, and the display/storage of data. Although these topics may be introduced as part of a final year project, we believe that a dedicated course focused on interfacing and modern measurement techniques should form a central component of an experimental physicist's training.

A number of issues naturally arise regarding the design and development of such a course. First, ideally the material covered should not only encompass topics related to interfacing, instrumentation, and programming but also strengthen the students' understanding of physics concepts covered in lecture modules taken in parallel with the laboratory course (or introduce new concepts in a manner not possible in a conventional lecture). This leads to considerable opportunities for combining computer-aided learning with ''real world'' experiments. Second, aptitudes for computer programming vary strongly across a group of students. Therefore the computer language chosen for the course should be as user-friendly and intuitive as possible so that difficult programming concepts do not obscure the interfacing techniques and physics underlying a particular experiment. Third, the programming language should preferably be widely used in both academia and industry so that the students will have acquired an important transferable skill on completing the course. Finally, there should be significant scope for the development of transferable skills through both oral and written presentation.

National Instruments' LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is an industry-standard programming language.[1] While this is important in terms of training students in state-of-the-art technology, from a pedagogical perspective LabVIEW also has the significant advantage that it is a graphical programming environment. Programs are written by ''wiring'' icons together using standard point-and-click mouse techniques—quite unlike the purely text-based programming associated with conventional languages such as *C* or FORTRAN. In our experience, students who find conventional computing modules rather conceptually difficult are significantly more at ease with LabVIEW and can create functioning programs in a fraction of the time it would take with text-based languages.

An additional important feature of LabVIEW is the availability of a comprehensive set of easy-to-use functions dedicated to signal acquisition, processing, and analysis. This enables the development of a series of interactive programs [or, in LabVIEW parlance, virtual instruments (VIs)] which guide the student step by step through important areas such as Fourier analysis, filtering, phase sensitive detection, and process control. As the students are required to first construct the VIs they use, this has the potential to engender a deep understanding of the physics and mathematics underlying a particular topic.

In the following sections we detail the LabVIEW-based *Interfacing and Modern Measurement Techniques* undergraduate (second year) laboratory course that has been running in Nottingham since 1998. Although a text describing the use of LabVIEW in the undergraduate laboratory has recently been published,[2] the development of the course described below predates the publication of that book, and, as such, the present paper covers a range of LabVIEW topics and experiments that are either complementary to those covered in Ref. 2 or, in the majority of cases, have not been previously described in an undergraduate laboratory context. Furthermore, a defining feature of the Nottingham course is that the emphasis throughout is on physics experimentation and concepts. This represents a significant departure from earlier interfacing courses which necessarily focused on the details of both hardware and low level languages. It is also

Table I. Structure of the ''Interfacing and Modern Measurement Techniques'' (LabVIEW) course.

Week 1	Building virtual instruments: Fundamentals
Week 2	Digital and analog I/O
Week 3	Signal analysis and signal processing
Week 4	Electrical and mechanical resonance (Assessed laboratory notebook account)
Week 5	Process Control/GPIB communication
Week 6	Triggering: Measuring thermal diffusivity using a flash gun (assessed formal account)
Week 7	Project
Week 8	Project
Week 9	Project
Week 10	12 min project talk

worth noting that because the Nottingham course is now into its fifth year we have collated a considerable amount of formal and informal feedback from undergraduate students about the course.

II. COURSE OUTLINE

The *Interfacing and Modern Measurement Techniques* course forms a single semester component of the two semester (20 week) second year undergraduate laboratory module in the School of Physics & Astronomy. Prior to the course the students will have taken a first year experimental physics module which involves no computer-based experimentation other than the use of software for graph plotting and linear regression analysis. A module on fundamental C programming is taken by all first year physics students. Most, though not all, of the students will also have experience with rudimentary analog electronics through a first year laboratory module that runs in parallel with the electronics lecture course. Again, this does not involve computer-based experiments. In parallel with the *Interfacing and Modern Measurement Techniques* course second year students take core lecture modules in mathematical physics (where Fourier transforms are covered in considerable detail); electromagnetism; quantum physics; solid state physics (crystal structure, phonons, thermal properties of insulators); physical optics; thermal physics, and statistical mechanics.

The laboratory manual for the course contains a series of tutorials related to LabVIEW programming and details the tasks and experiments the students are expected to complete each week (see Table I). However, a great deal of attention was paid to ensuring that the manual was not wholly prescriptive—a considerable amount of the course material is either largely ''problem solving''-related or requires the students to make conceptual links with the material covered in their lecture courses (this is discussed in detail below). This means that the LabVIEW course necessitates pro-active instructor involvement to ensure that the students are considering the problem or task in an appropriate fashion. As shown in Table I, the students are not assessed on weeks 1–3 of the course [although it is stressed that a good understanding of the topics covered in the early stages of the course is essential for progression through the later (assessed) stages of the course]. The lack of assessment in weeks 1–3 is motivated by the requirement that students of widely varying ability take the course and is an attempt to ensure that the students are more concerned with gaining a deep understanding of the material rather than simply completing as many tasks as quickly as possible in an effort to improve their grades.

Approximately 20 students are involved in each LabVIEW laboratory session (of which there are three per week). Each student is equipped with a PC which houses a National Instruments AT-MIO-16E-10 data acquisition (DAQ) card. This card features 16 single ended (or 8 differential) analog-to-digital converter (ADC) channels with 12 bit resolution, channel-independent software-selectable gains, and 100 kHz sample rate. In addition, the AT-MIO-16E-10 has two analog output channels (again with 12 bit resolution) and eight digital (TTL) I/O lines. Connections to and from the DAQ card are made via National Instruments' multi-connector blocks housed within homebuilt die-cast ''breakout'' boxes. Seven of the PCs in the LabVIEW laboratory also house National Instruments GPIB (General Purpose Interface Bus) cards.

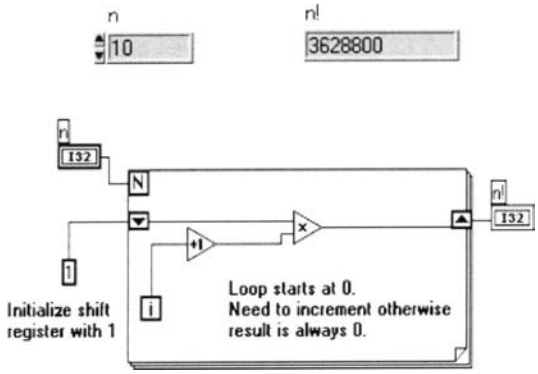
III. TASKS, EXPERIMENTS, AND PROJECTS

In the following sections we briefly describe the principal topics covered in the course. Examples of LabVIEW virtual instruments (VIs) are included throughout. Note that all VIs were written in LabVIEW Ver. 4.1. Although these VIs will run with no modification in more recent versions of LabVIEW (at the time of writing, LabVIEW 6.1 is used in our undergraduate modules), a *waveform* data-type has been introduced in LabVIEW 6.1 and provides a slightly more intuitive method of storing and displaying waveform data. We return to a discussion of waveform data in the following sections. Our choice of topics for the course was largely driven by two requirements: that the material was of key significance to computer-based experimentation and it easily lended itself to computer-aided teaching of core physical concepts.

A. Graphical programming, virtual instruments, and digital I/O

In the first week the students learn the fundamentals of ''G''—the graphical programming language at the heart of LabVIEW—and write simple VIs that involve little interfacing or data acquisition. Although the lab manual[3] contains a variety of basic graphical programming tutorials we will not cover the details of ''G'' here—see Refs. 1 and 2 for excellent introductions to the language. The programs the students construct on week 1 include a simple calculator and a VI that calculates *n*! (Fig. 1). Loops are a prerequisite for the latter VI and these, along with case structures (the LabVIEW equivalent of *if-then-else* blocks) and sequence structures are introduced early in the course.

Sequences are a programming element specific to LabVIEW and exemplify the ''data flow'' approach to coding that is inherent to the ''G'' language. We find that for those students (and instructors!) with considerable experience of procedural languages, the necessity to consider ''data flow,'' which may sometimes appear counter-intuitive and ''nonlinear,'' through a block diagram can sometimes be an initial stumbling block in developing efficient LabVIEW programs. However, as with any other programming language, what at first seems counter-intuitive becomes almost second nature given sufficient practice. From the instructor's point of view, an additional important advantage of LabVIEW's graphical interface is that generally a student's code is somewhat easier to debug than for conventional procedural languages. The importance of breaking LabVIEW block diagrams into a



Loop starts at 0.
Need to increment otherwise result is always 0.

Fig. 1. A simple LabVIEW virtual instrument to calculate *n* factorial. The front panel in this case consists of one control (n) and one indicator ($n!$). Construction of a VI similar to this is one of the first tasks the students complete.

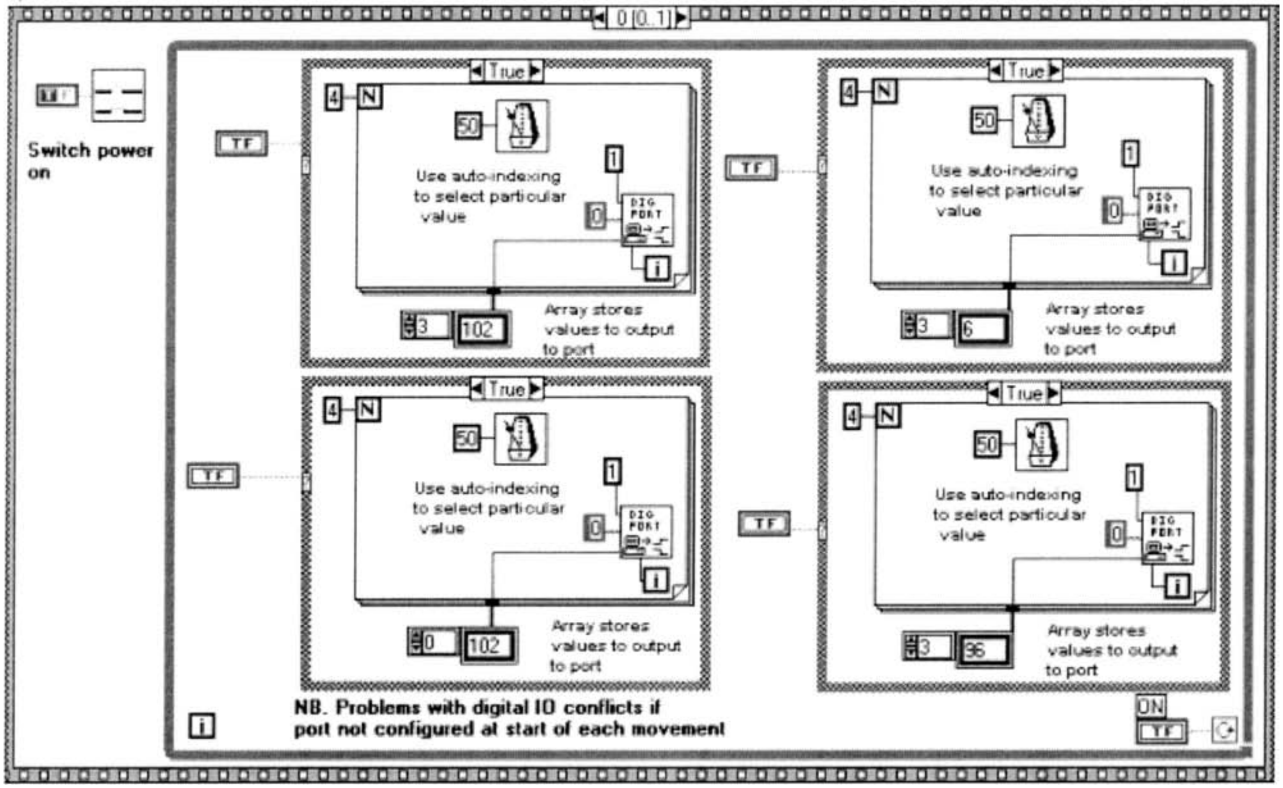
modular form via the use of ‘‘sub-VIs’’ (see, for example, Ref. 1) is stressed to the students early in the course.

As an introduction to interfacing with LabVIEW, the students build a simple program to light a series of LEDs via a digital port on the AT-MIO-16E-10 DAQ card. By the end of the first 6 h session, the majority of students are sufficiently proficient with LabVIEW to construct a VI that drives the stepper motors of a robot ‘‘buggy’’ similarly interfaced via a digital port (Fig. 2). They are expected simply to write the digital I/O LabVIEW code that drives the buggy—all peripheral electronics (and the buggies themselves) are ready to use and require no construction or modification by students.

B. Sampling, Fourier analysis, and signal generation

The students are introduced to analog-to-digital (AD) conversion at the start of the second week of the course. Starting with a rudimentary data acquisition program, they progress to using a range of standard LabVIEW analog input VIs to acquire and output waveforms at various sample rates on more than one AD converter channel (Fig. 3). Signals are acquired from basic analog signal generators. An important feature of the VI shown in Fig. 3 is that it permits *simultaneous* acquisition and generation of signals—a key requirement for a number of VIs that the students construct later in the course. Furthermore, the construction of Fig. 3 requires that the students understand the importance of LabVIEW clusters (discussed below) in scaling the *x*-axis of any waveform graph they plot. Without inclusion of a cluster, the *x*-axis of a LabVIEW graph may represent sample number as opposed to a more physically significant variable such as time or frequency. We find that this point—the importance of including physical units for any measurement made with LabVIEW—must be stressed continually throughout the course.

In more recent versions of LabVIEW (from version 6.1 onwards), the default representation of waveform data is a LabVIEW data structure called a cluster which consists of a value for the origin of the time ‘‘axis’’ (t_0), a value for the spacing between data points (i.e., Δt), and the data values themselves. This cluster is termed a *waveform* data type. Although the introduction of the waveform data type in LabVIEW 6.1 is particularly useful in terms of facilitating the rapid construction of signal processing VIs for academic research and industrial applications, from a pedagogical per-



Use auto-indexing to select particular value

Array stores values to output to port

NB. Problems with digital IO conflicts if port not configured at start of each movement

Fig. 2. A LabVIEW program used to control a ‘‘robot’’ buggy. Most students will have the ability to successfully complete a similar VI by the end of the first 6 h session of the course.

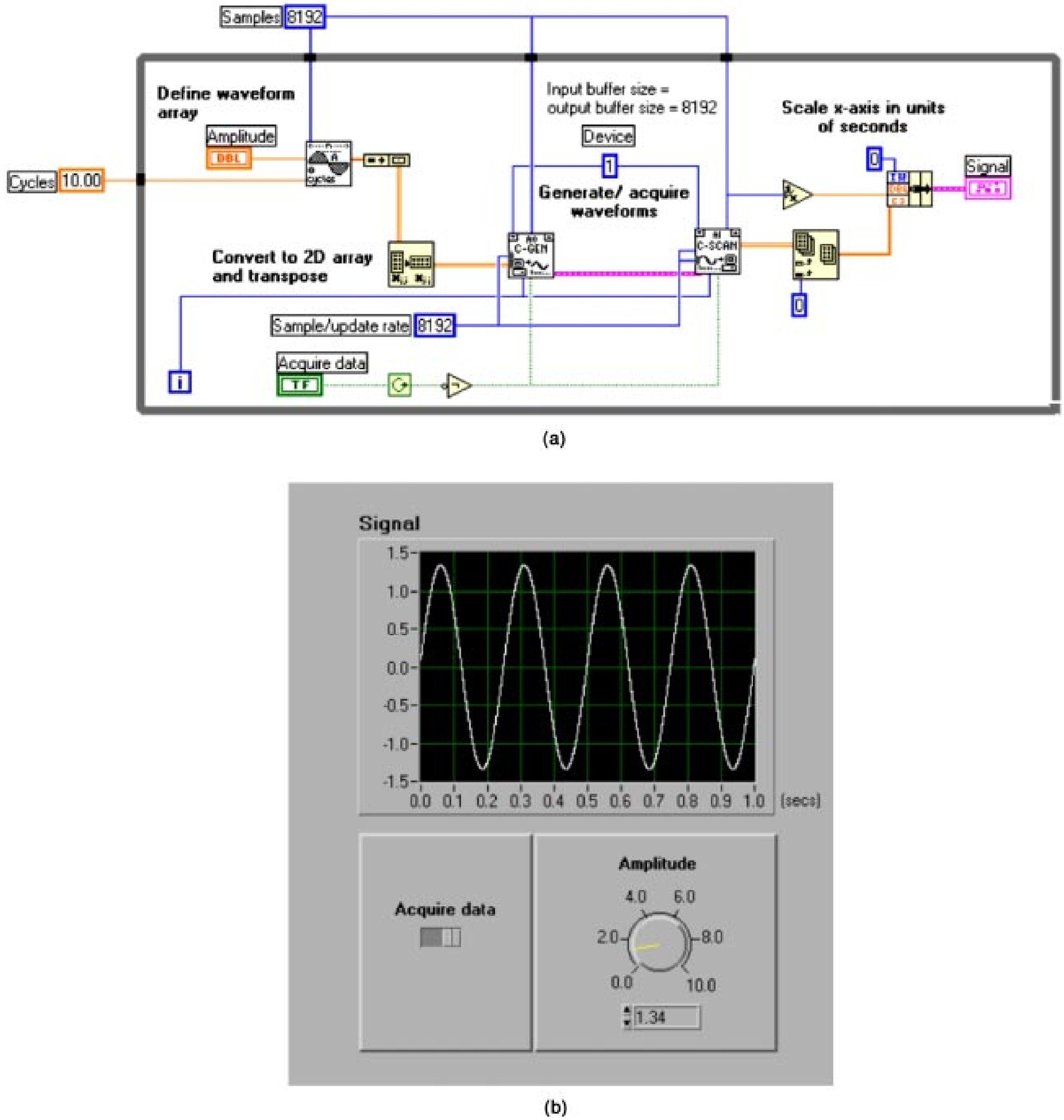
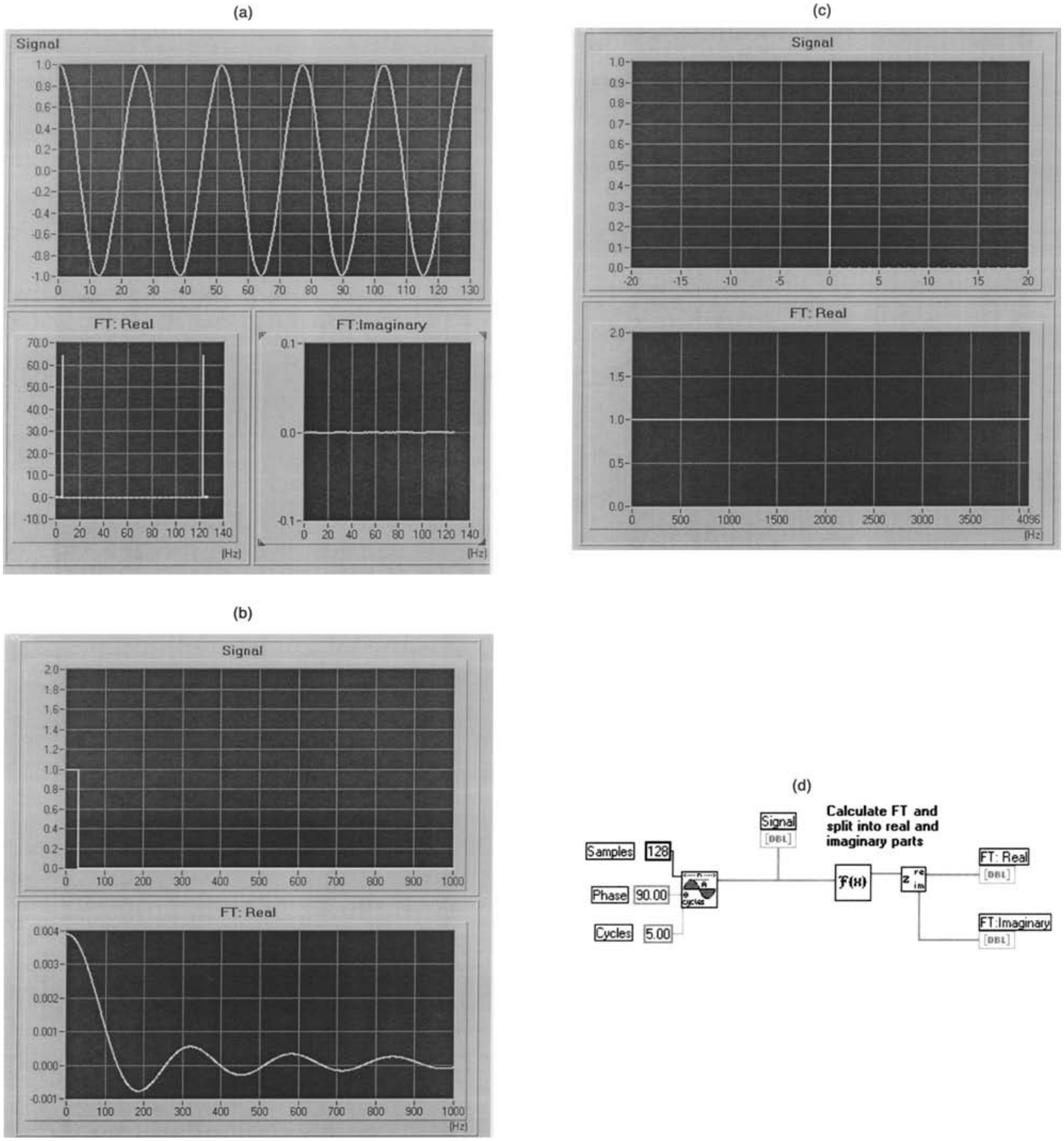


Fig. 3. (a) Block diagram of a VI to simultaneously acquire and generate waveform data. VIs similar to this are at the heart of the students' use of LabVIEW to study resonance phenomena. (b) Front panel corresponding to block diagram shown in (a).

spective it does tend to obscure the relationship between clusters and waveform data scaling that is a feature of the VI shown in Fig. 3. Hence, in the Nottingham course, considerable effort is spent on highlighting that the *waveform* data type is simply a particular type of cluster.

LabVIEW is a powerful signal processing tool with a wide variety of VIs enabling the calculation of functions such as fast Fourier transforms, power spectra, and correlation functions. Student feedback indicates that the ability to display *in real time* the Fourier transform or power spectrum for a waveform considerably strengthens their understanding of Fourier analysis and leads to a better appreciation of its importance in physics. An important distinction must however be drawn between analytical FTs and the discrete FTs calculated by LabVIEW (or any computer package). A considerable amount of time is therefore devoted to the discrete FT and related subjects such as aliasing before the students begin to determine FTs for various waveforms. Although aliasing is a somewhat ‘‘difficult’’ concept for undergraduate students to immediately appreciate on first encountering the topic, we feel strongly that it must be included in any course on modern measurement techniques given its key importance in digital data acquisition and signal processing.

Some examples of the tasks set in the Fourier analysis section of the Nottingham LabVIEW course are shown in Fig. 4. Following a discussion of the discrete FT and double-sided/single-sided FT representations, the students are asked



Samples 128

Phase 90.00

Cycles 5.00

Fig. 4. Simple LabVIEW VIs to illustrate key concepts in Fourier analysis. (a) Real and imaginary parts of Fourier transform of pure sine waveform; (b) illustration of the sinc form of the FT of a ''top hat'' or pulse function; (c) digital approximation to a delta function—a single sample impulse—and its corresponding FT; (d) block diagram corresponding to front panel shown in (a); very minor changes to this block diagram were used to produce (b) and (c).

to write a VI that displays the real and imaginary parts of a pure sinusoid waveform [Fig. 4(a)]. Simple questions related to the symmetry of the waveform and the FT, the reasons underlying the absence of real or imaginary parts of the FT, and the positions of the peaks in the FT are posed in the lab manual to probe students' knowledge and understanding of both discrete and analytical FTs. The absence of negative frequencies in LabVIEW's representation of the FT and the related presence of peaks *above* the Nyquist limit require careful explanation by the instructors in the course.

As noted above, an important aspect of the Fourier analysis component of our LabVIEW course is to test and strengthen understanding of a number of fundamental relationships between time and frequency space. The students start by varying the duty cycle of a square wave, exploring the changes that occur in the discrete FT as the signal progressively approaches the limit of a single aperiodic pulse (i.e., a 'top hat' function). Thus, as shown in Fig. 4(b), the students observe the evolution of frequency space from the discrete set of odd-numbered harmonics expected for the Fourier series representation of a square wave to the sinc function they have encountered in their lecture modules as the FT of a top hat function. The distinctions between analytic and discrete FTs again need to be highlighted.

From an examination of the width of a pulse waveform (created using a standard LabVIEW ''Signal Generation'' VI) and the width of the region of its FT between dc and the first zero crossing [Fig. 4(c)], the students are then prompted to explore the effects of signal duration on frequency bandwidth. Although not explicitly covered in our LabVIEW course, this in turn could easily lead to a discussion of a very wide range of topics including the uncertainty principle, wavepackets, and diffraction phenomena. The frequency content of a pulse which is a single sample wide (i.e., a digital approximation to a delta function) and that of a dc signal are determined in quick succession to strengthen the students' understanding of the fundamental relationship between the time and frequency domains [Fig. 4(d)]. The students are asked to compare the output of the VI shown in Fig. 4(d) with the result of the expression for the discrete FT:

$$X_k = \frac{\sum_{n=0}^{N-1} x_n \exp(-2\pi i k n/N)}{N}, \tag{1}$$

where X_k are the data values corresponding to the Fourier transform of the input data x_n and N is the total number of samples.

The remaining parts of the lab session are devoted to two key topics in digital signal processing: power spectra and spectral leakage. The ability to recover spectral components via windowing is demonstrated. This again places an important emphasis on the high spectral bandwidth associated with time-limited signal discontinuities. However, the use of LabVIEW to illustrate aliasing and windowing phenomena has been elegantly discussed by Chugani, Samant, and Cerna[4] and is thus not covered here. Finally, the students are asked to construct a VI that enables simple spectral analysis of speech samples they record.

C. Electrical and mechanical resonance

The first assessed laboratory session involves the development of VIs to measure the resonance characteristics of simple mechanical and electrical systems. Not only does this build on the knowledge of resonant systems gained in a Vibrations and Waves module taken in the first year of the degree course, it also extends the application of Fourier analysis covered on week 2. The students are first expected to construct a VI that varies the frequency of a sinusoidally varying ac voltage applied to a simple LRC circuit and simultaneously measures the current flowing in the circuit (by monitoring the voltage across a resistor). This requires some relatively simple modifications to the block diagram shown in Fig. 3.

We find that construction of the VI to measure the LRC resonance curve provides a thorough test of a student's understanding of many elements of LabVIEW including ''for'' loops—it is not uncommon for there to be a degree of confusion between the ''N'' (total number of iterations) and ''i'' (iteration counter) icons on the block diagram. Similarly, the importance of clusters in providing correct scaling of the *x*-axis is again emphasized. Having generated a resonance curve for the LRC circuit by varying the frequency in fixed increments, the students (following appropriate prompting by the lab manual and the instructors) are asked to build a VI that applies a sharp impulse to the circuit and simultaneously measures its response. The FT of the exponentially decaying sinusoid response produces the Lorentzian resonance curve (Fig. 5).

From their study of FT's in week 3 of the course (and in their lecture modules), the students are asked to explain why a FT of the output of the circuit produces the resonance curve. Although a common problem posed in lecture modules on Fourier transforms is the analytic calculation of the FT of a function of type $f(t) = e^{-\lambda t} e^{i\omega t}$, we find it is sometimes difficult for undergraduates to see the physics underlying the evaluation of the integrals. Combining the LabVIEW procedure illustrated in Fig. 5 with an analytical approach to determining the FT leads to much greater physical insight into resonance phenomena.

The VI in Fig. 5 is then applied directly to the measurement of resonance in a mechanical system. Turvey[5] has described a simple undergraduate experiment to measure Young's modulus for steel via the detection of the resonance frequency for a cantilever driven by a sinusoidally varying magnetic field. We have adopted this experiment for the resonance section of our course, but instead of detecting the resonance frequency for a given length of steel cantilever ''by eye'' (as discussed by Turvey[5]) we use the VI of Fig. 5 combined with the experimental set-up shown in Fig. 6 to determine resonant frequencies. The LabVIEW generated impulse is amplified by a power transistor circuit and applied to a solenoid which in turn generates a transient motion of the steel cantilever. The motion is detected by an inexpensive piezoelectric crystal glued onto the steel strip.

A typical resonance spectrum and a plot of resonant frequency versus $1/l^2$ are shown in Figs. 7(a) and 7(b), respectively. What is instructive for the students is that attempting to determine the resonant frequency from the ''raw'' time domain piezoelectric data [upper graph in Fig. 7(a)] is sometimes difficult due to the presence of a large amount of 50 Hz pickup. The pickup arises because the connections to the piezoelectric are unscreened soldered joints. However, despite the high levels of interference it is straightforward to isolate the resonant peak from the 50 Hz noise using a FT—a valuable lesson in the importance of Fourier analysis. Note, however, that the importance of using differential inputs and screened connections to measure low level signals is of course stressed throughout the course. Using the theory outlined by Turvey,[5] the fundamental eigenmode of the steel rule is associated with a frequency f_1 given by

$$f_1 = 0.5596\sqrt{\frac{EI}{\mu l^4}}, \tag{2}$$

where E is the Young's modulus, I is the second moment of area of the steel rule, μ is the mass per unit length, and l is the free length of the rule.

The students can thus use a graph similar to that shown in Fig. 7(b) to determine the Young's modulus for steel. The data shown yield a value of 190 (±10) GPa—well within the range of values noted in Ref. 5. At this point we note that a significant limitation of LabVIEW with regard to its use in an undergraduate laboratory course is the inability to include error bars on graphs. Thus, although data and spectra may be acquired and displayed with ease using LabVIEW, students in the course are advised to use a graphing package other than LabVIEW to plot data for laboratory and project reports when error bars must be included. Simple to use ''Write to

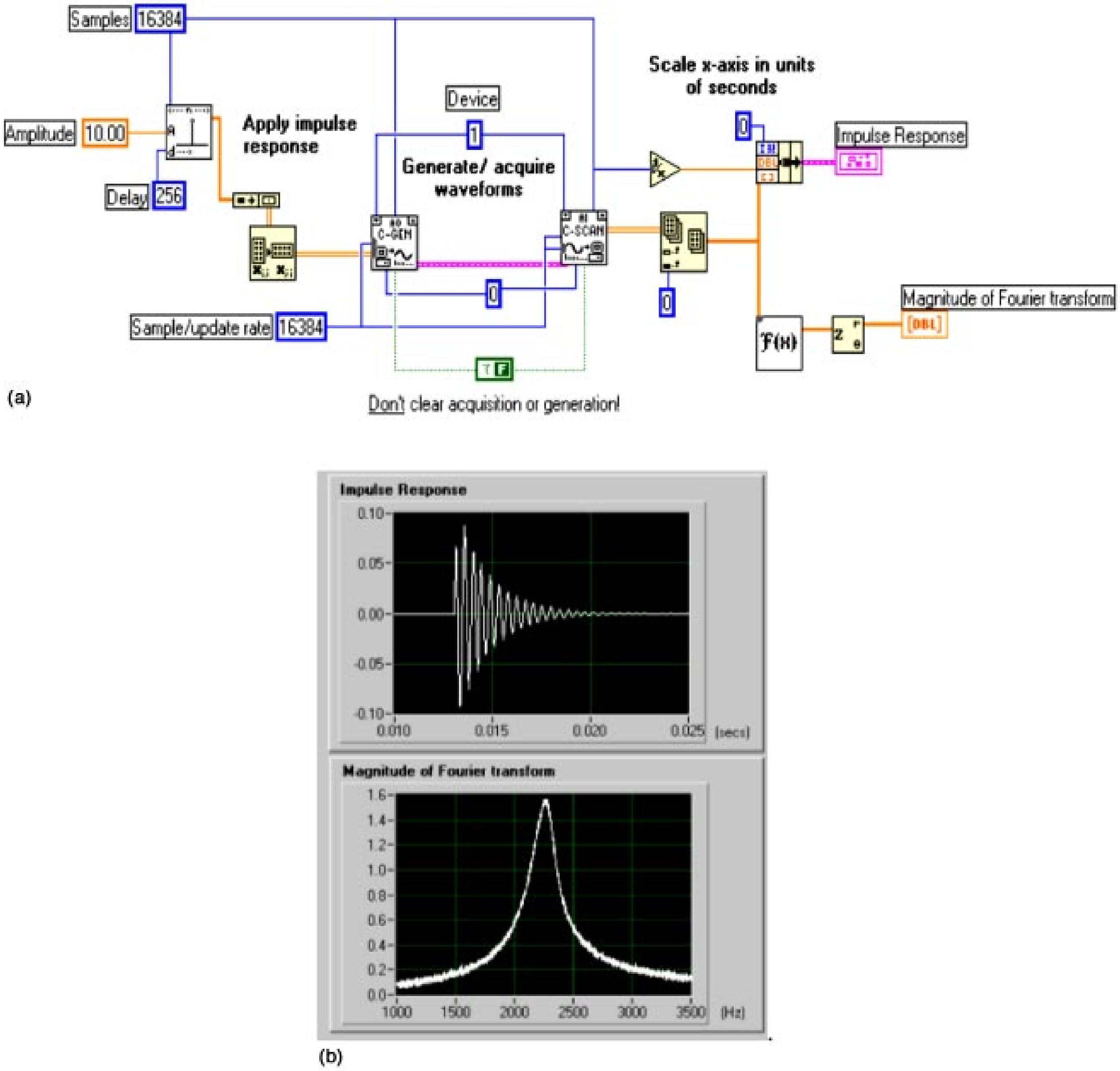


Fig. 5. Front panel and block diagram of VI used to produce a resonance curve for a series LCR circuit. The block diagram involves relatively minor modifications to the VI shown in Fig. 3.

Spreadsheet'' VIs are supplied with LabVIEW to facilitate transfer of data to and from graph plotting packages, spreadsheets, and word processing applications.

D. Noise, filtering and phase sensitive detection

Initially, a simple noise generator circuit or a LabVIEW noise generation VI is used to produce a Gaussian-distributed set of data values. Using functions available on LabVIEW's ''Probability and Statistics'' palette the students determine the mean and variance of the data and examine the effect of increasing the number of samples used to form the Gaussian curve. They are prompted to interpret these measurements in terms of the error analysis techniques covered in their first year laboratory sessions.

The key concepts underlying this laboratory session are filtering and noise reduction. In particular, the concept of reducing or modifying the measurement bandwidth is a recurring theme. Hence, low-, high- and band-pass filters are initially discussed and their operation demonstrated using the standard LabVIEW Butterworth Filter VI. The concept of impulse response and its connection to the output of a filter through the convolution process are discussed and illustrated with suitable examples.

A signal averaging VI is simple to implement using LabVIEW (Fig. 8). The construction of this VI forms an important task in this lab session and we find that the students are generally impressed by the level of noise reduction that is possible simply by summing the signal over time. Two aspects of signal averaging need careful explanation: first, the signal-to-noise ratio improves as n (where n is the number of iterations) and, second, in this case the averaging process filters by effectively narrowing the measurement bandwidth.

Foremost amongst filtering and signal recovery techniques is, of course, lock-in amplification. This sometimes can be a difficult topic for undergraduate physics students to grasp

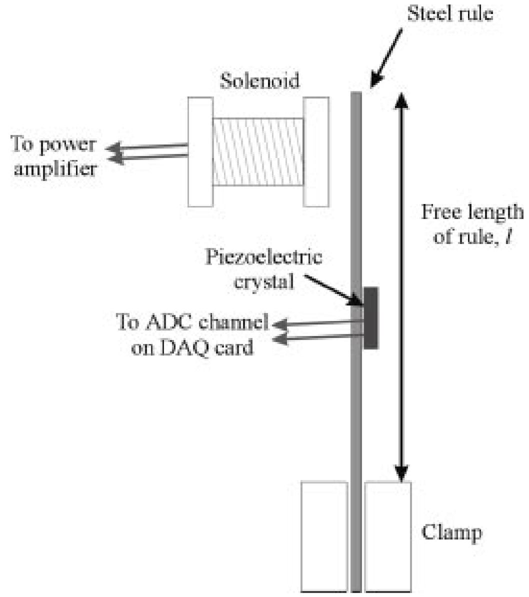


Fig. 6. Experimental apparatus used to determine Young's modulus for steel via measurements of the resonant frequency of a cantilever.

largely because conventional descriptions of lock-in amplification tend to obscure the correlation function aspects of the filtering process with details of the analog electronics at the core of the amplifier (although the use of digital lock-in amplifiers is increasingly common). With LabVIEW it is possible to directly ''code'' the correlation integral that underlies lock-in signal processing into a virtual instrument (see Fig. 9).

The students are asked to construct the lock-in VI from a general description of phase sensitive detection and an explanation of the correlation integral on the right hand side of

$$\frac{V_R \otimes V_S}{nT} = \frac{1}{nT}\int_0^{nT} V_1 \sin(\omega t) V_2 \sin(\omega t + \delta) dt, \qquad (3)$$

where $V_S = V_1 \sin(\omega t)$ and $V_R = V_2 \sin(\omega t + \delta)$ are the sample and reference waveforms, respectively. Although most students find this task initially quite daunting, the only element that requires a relatively large degree of effort is to ensure that the integration is carried out over the correct (integer) number of cycles [i.e., to ensure that the limits of integration in Eq. (3) are applied appropriately]. In the block diagram shown in Fig. 9 an ''Array Subset'' node is used to select the appropriate number of samples corresponding to n periods (nT) (where T = total number of samples/number of cycles).

Following construction of the lock-in VI, the effects of a number of important parameters are studied. First, nT is varied and changes in signal to noise ratio are monitored. For those students who have used an analog lock-in amplifier in previous laboratory courses, it is important to stress the relationship between the ''number of cycles'' control in Fig. 9 and the time constant of a conventional lock-in. The band pass properties of the LabVIEW lock-in are explored by varying the reference frequency and recording the output of the VI. Finally, phase sensitive detection is introduced through an analysis of the changes in the output of the lock-in VI produced via variation of the phase control [the students are told (and asked to confirm) that evaluating the integral on the right hand side of Eq. (1) yields the result $(V_1 V_2/2)\cos\delta$ (assuming identical signal and reference frequencies)].

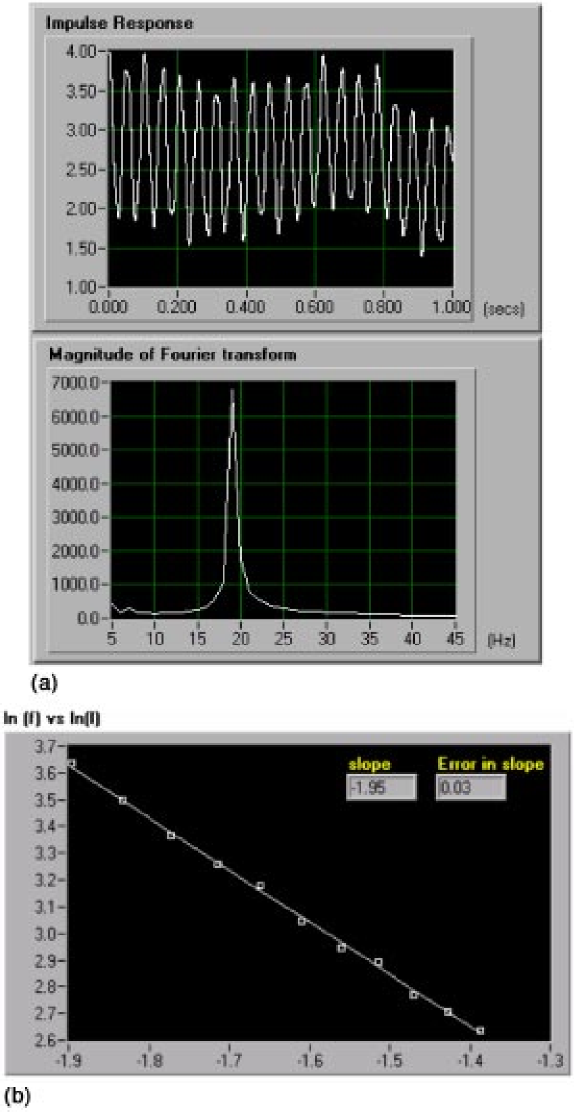
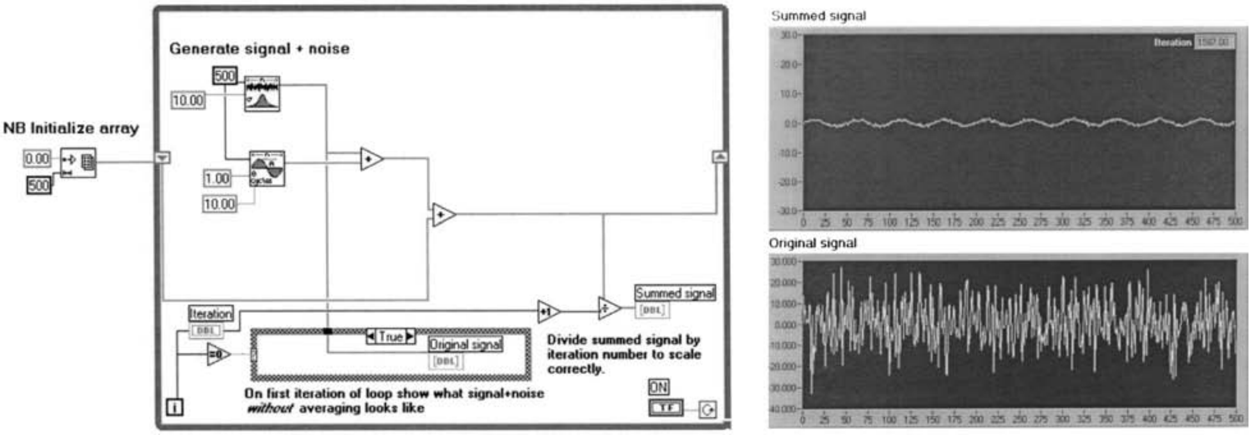


Fig. 7. (a) Signal from piezoelectric crystal following impulse excitation of steel rule. The upper graph is the time domain response, the lower graph the corresponding Fourier transform. (b) Plot of the cantilever resonant frequency against $1/l^2$ from which a value for the Young's modulus for steel may be determined (see text for details).

E. Process control and GPIB communication

Although a detailed discussion of process control theory (including, for example, topics such as Bode plots, poles, and Laplace/Z transforms) is well beyond the scope of the course, the basic concepts underlying PI (proportional-integral) control are presented to the students. (See Ref. 4 for an excellent discussion of the use of LabVIEW for advanced process control applications. Note also that PI control is discussed in some detail in Ref. 2.) Students are first expected to construct a simple proportional controller (Fig. 10) to control the temperature of a heater circuit. A key observation is that proportional gain alone leads to a steady state error. The modifications necessary to produce a PI controller (via the use of a shift register to store the integral correction term



Divide summed signal by iteration number to scale correctly.

On first iteration of loop show what signal+noise *without* averaging looks like

Fig. 8. A LabVIEW signal averaging VI.

from iteration to iteration) are described and, on adding the integral term, the students find they can eliminate the steady state error.

The second part of the lab session focuses on General Purpose Interface Bus (GPIB or IEEE) control. A number of simple VIs to send instructions to, and read measurements from, an IEEE-compatible multimeter are first constructed by the students and saved as sub VIs. The sub VIs are in turn used in the construction of a LabVIEW program to measure the I-V characteristic of a Si diode and to subsequently calculate the value of the ideality factor (η) from

$$I = I_0 \exp\left(\frac{eV}{\eta kT}\right). \tag{4}$$

For the earliest version of the course, the students were also expected to vary the temperature of the diode over a range of 30K in 5K increments (using the PI controller constructed in the first half of the lab session), measuring an I-V curve at each temperature. From the variation of I_0 with increasing temperature the value of the band gap in silicon can be determined (as I_0 provides a measure of the number of minority carriers and is therefore in turn related, through a simple Boltzmann factor, to the band gap energy). Unfortunately, it was found that the vast majority of students found this task impossible to complete within the 6 h lab session and it was thus removed from later versions of the course. Nevertheless, the task remains an interesting and challenging method of teaching a combination of PI control and GPIB communication.

F. Thermal diffusivity and triggering

The second assessed element of the course is concerned with the use of digital triggering techniques in the measurement of the thermal diffusivity of Cu. We follow the experimental procedure outlined initially by Parker *et al.*,[6] measuring the characteristic time associated with the temperature rise at the face of a Cu disc heated with a high intensity light pulse (from a conventional camera flash gun). A number of Cu discs of different thicknesses (in the range 1–2.5 mm) are used. A schematic diagram of the experimental setup is shown in Fig. 11. The AI Waveform Scan VI is used to collect data when triggered via the Dig. Trig. A input to the AT-MIO-16E-10 DAQ card. The trigger signal (a TTL pulse) is activated when the flash gun is fired. The signals from thermocouples attached to the front and back faces of each Cu disc are amplified and measured via differential ADC channels.

The rise in temperature of the back of the disc measured directly following a flash gun heat pulse is shown in Fig. 12. Although it is possible to carry out nonlinear least squares fitting within LabVIEW (using the Levenberg-Marquadt algorithm[3]) and thereby determine the thermal diffusivity coefficient from an appropriate fit to the data shown in Fig. 12, to date students in the course have employed the method of data analysis discussed by Parker *et al.*[6] and Rogers.[7] This involves plotting the time taken for the back of the disc to reach half its maximum temperature against the square of the thickness of the disc. The slope of this graph is $0.139/\alpha$ where α is the thermal diffusivity.[6,7]

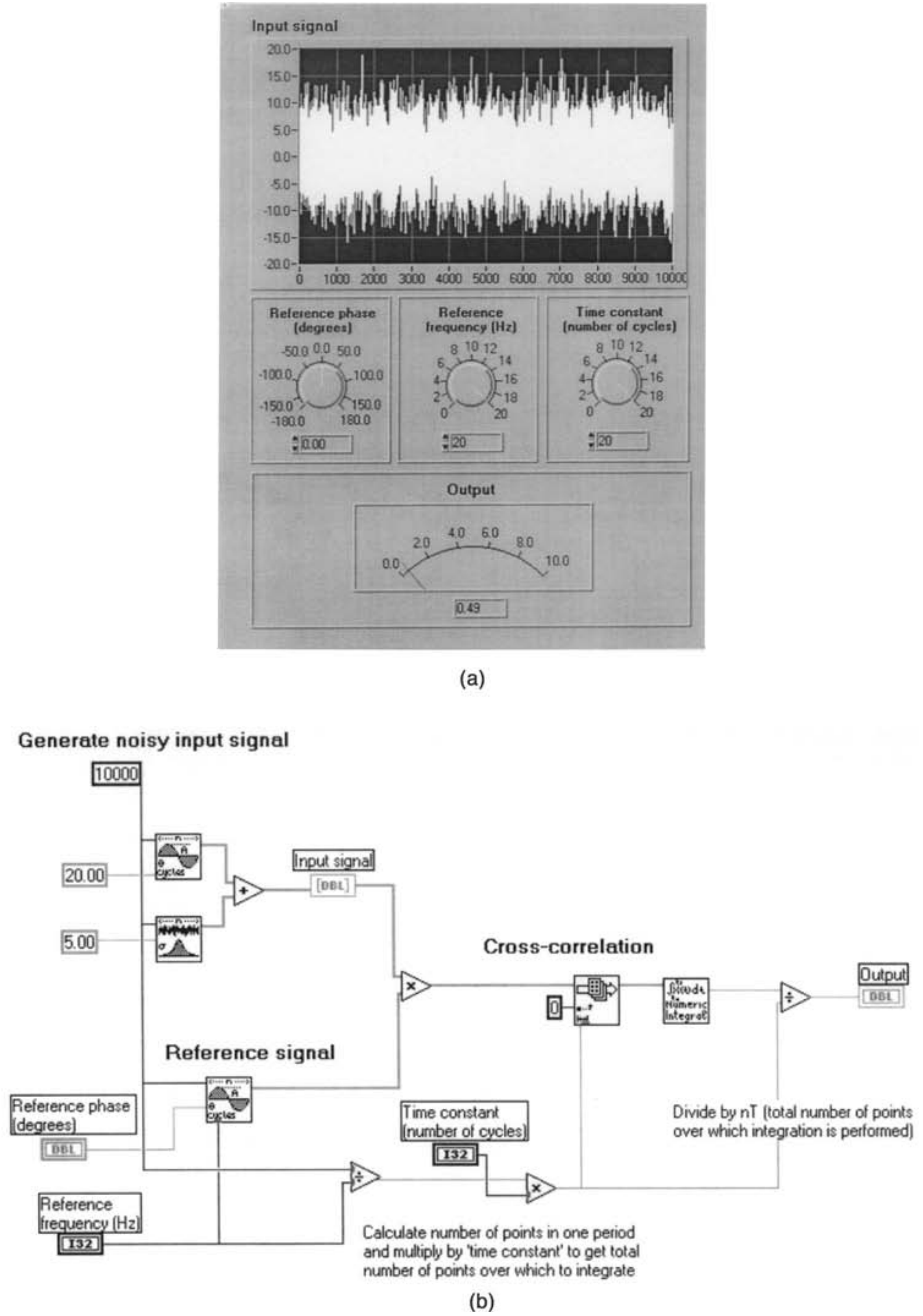
In addition to illustrating the importance of triggering techniques, this experiment clearly demonstrates the clear advantages associated with computer-based experimentation where large data sets can be acquired and analyzed with ease.

G. Projects

During weeks 7–9 of the course the students complete a project (in pairs). For each project a single page describing the aims and objectives, with appropriate references to textbooks or scientific papers, is initially handed out. From this, the students are expected to develop LabVIEW code that automates a particular series of measurements, interpret and analyze the resulting experimental data, and produce a 5000-word report in the style of a journal paper.

To date we have set five different projects (generally two pairs of students will carry out the same project in a given semester), namely: chaos in a ‘‘bouncing ball’’ electronic circuit; optical absorption in semiconductors; measurement of Planck’s constant via the photoelectric effect; hysteresis curves in ferromagnetic materials; and the Hall effect in metal-oxide-semiconductor field effect transistors (MOSFETs). We will briefly discuss just two of these—the chaotic bouncing ball and optical absorption projects—highlighting the advantages of using LabVIEW in each case.

In the initial stages of course development it was felt that the second year students should be introduced to chaos in nonlinear systems, largely to highlight that the analytical mathematics covered in other lecture modules is insufficient



Divide by nT (total number of points over which integration is performed)

Calculate number of points in one period and multiply by 'time constant' to get total number of points over which to integrate

Fig. 9. A LabVIEW "virtual lock-in amplifier." Concepts such as phase sensitive detection, the exceptionally narrow bandpass associated with lock-in amplification, and the dependence of signal-to-noise ratio on the lock-in "time constant" may be explored with simple VIs of this type.

to solve a very wide range of dynamical problems. One of the simplest examples of a system that exhibits chaotic behavior is a ball bouncing on a harmonically vibrating table. An electronic analog of this system, described in detail by Zimmerman, Celaschi, and Neto,[8] is used. There is one input to the circuit, an ac voltage which represents the table excitation waveform, and two outputs—voltages representing the ball position and velocity. The students are expected to write LabVIEW virtual instruments that drive the circuit and simultaneously plot not only the position and velocity waveforms but also phase space maps, Poincare sections, and bifurcation diagrams for the system (see Fig. 13). Data recording, storage, and analysis are facilitated greatly by LabVIEW. Previous experiments based on this circuit utilized a storage oscilloscope and photographs of the scope display to probe the rich dynamics of the bouncing ball.

Measurement of the optical absorption characteristics of a semiconductor is an important experiment to illustrate the key role band structure plays in determining the properties of condensed matter. A number of authors have previously described undergraduate experiments in this area.[9,10] For the LabVIEW course, the students are supplied with a grating monchromator, a white light source, a photodiode, a semiconductor sample, an analog lock-in amplifier, and appropriate lenses. They are expected to write a VI that scans the monochromator (via a stepper motor) while simultaneously measuring the output from the lock-in amplifier. Typical data for a GaAs sample are shown in Fig. 14. In addition to GaAs,

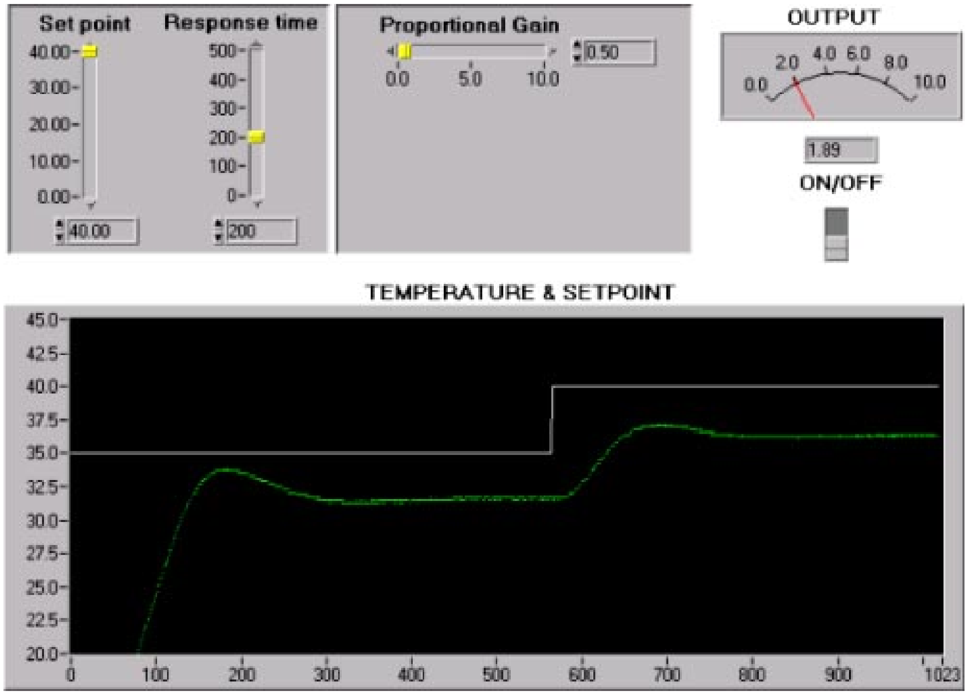


Fig. 10. Front panel of proportional control VI used to maintain a simple (resistive) heater at a constant temperature. Students learn that the steady state error (offset of the measured temperature from the set-point temperature seen in the graph above) may be removed using proportional-integral (PI) control.

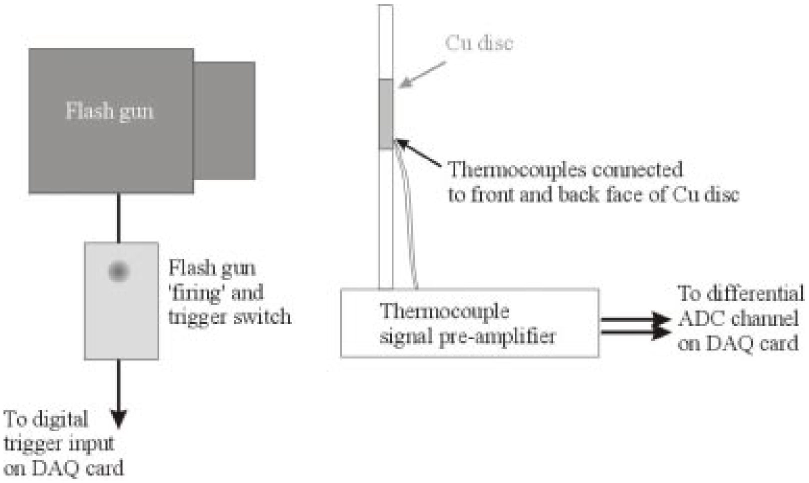


Fig. 11. Schematic diagram of the experimental apparatus used in determining the thermal diffusivity of Cu.

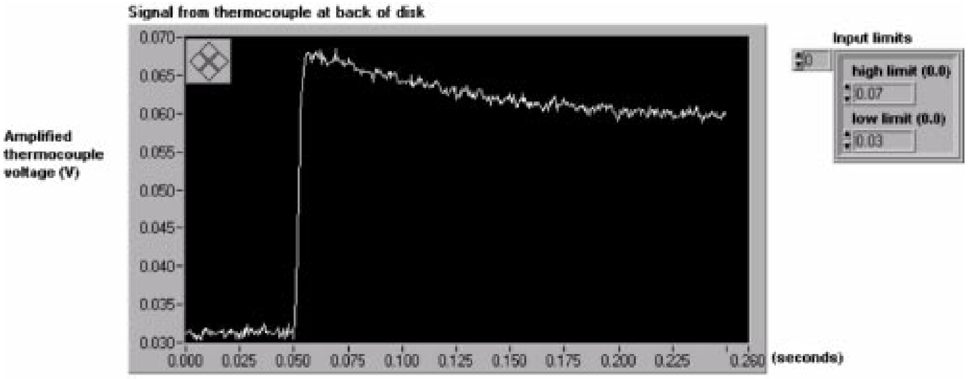
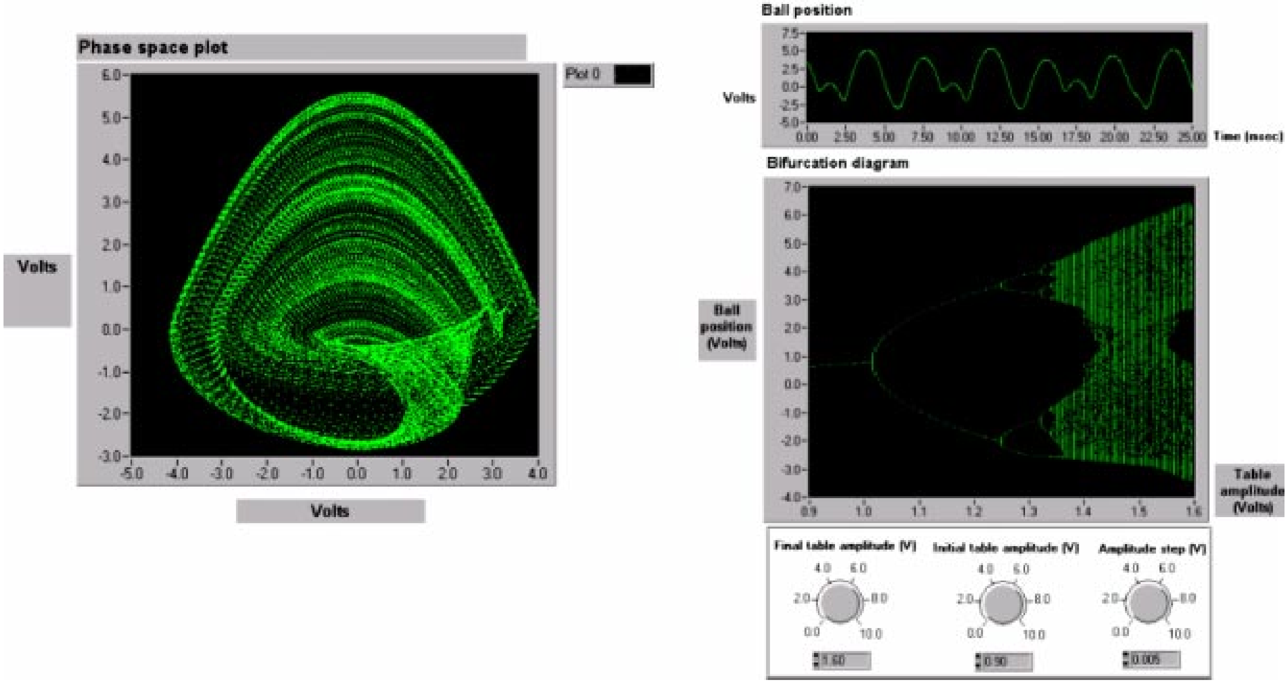


Fig. 12. Sample data from a flash gun experiment to determine the thermal diffusivity of Cu. Measurements of the rise in temperature (or, in this case, thermocouple voltage) as a function of sample thickness enable the value of α, the thermal diffusivity, to be calculated (see text for details).

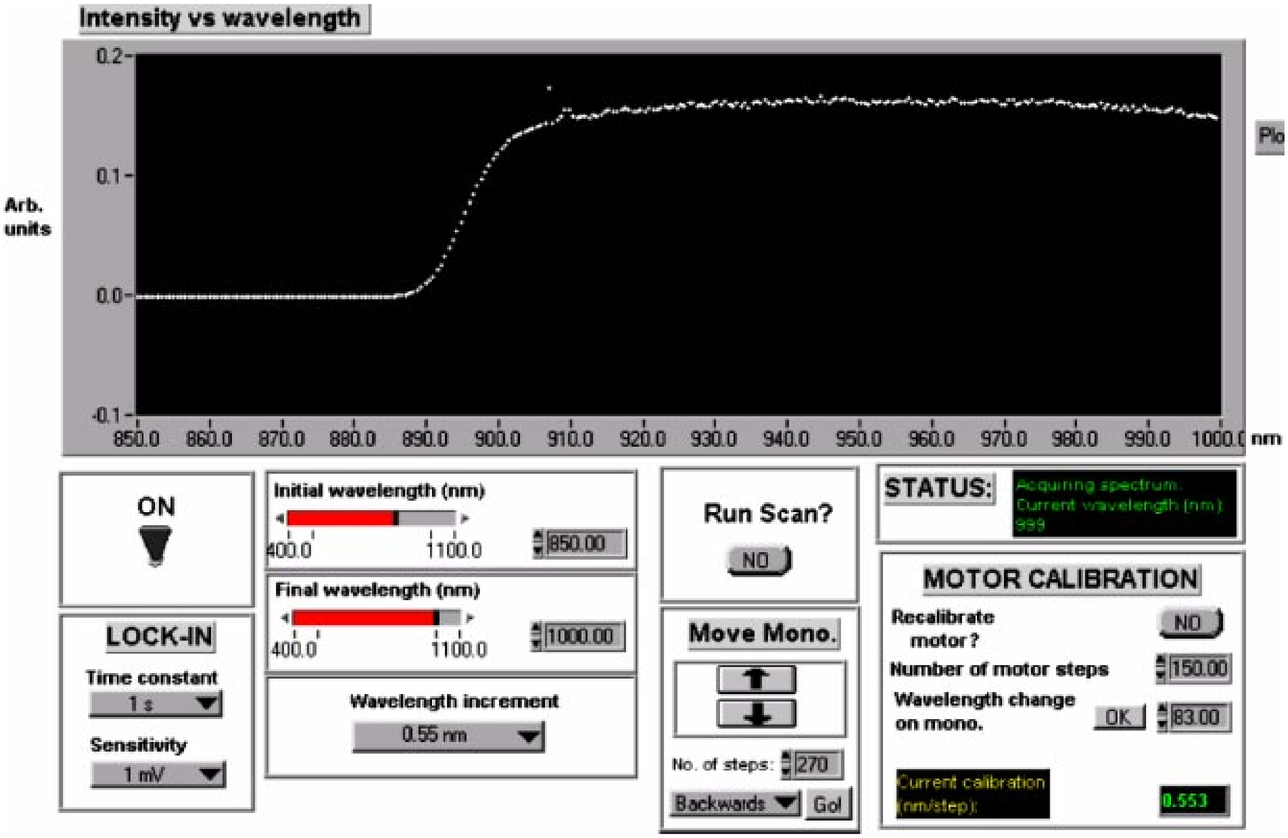


Final table amplitude (V) 1.60

Initial table amplitude (V) 0.90

Amplitude step (V) 0.005

Fig. 13. Bifurcation diagram (right) and strange attractor (above) for a ‘‘bouncing ball’’ electronic circuit. Construction of a VI to plot either a bifurcation diagram or a Poincaré section for a chaotic system is a nontrivial exercise for the students on the course.



Initial wavelength (nm) 850.00

Final wavelength (nm) 1000.00

Wavelength increment 0.55 nm

LOCK-IN

Time constant 1 s

Sensitivity 1 mV

Run Scan? NO

No. of steps: 270

Backwards

STATUS: Acquiring spectrum. Current wavelength (nm): 999

MOTOR CALIBRATION

Recalibrate motor? NO

Number of motor steps 150.00

Wavelength change on mono. 83.00

Current calibration (nm/step): 0.553

Fig. 14. Front panel of a LabVIEW VI used to measure the optical absorption characteristics of a semiconductor sample. The VI drives a monochromator via a stepper motor while simultaneously recording the output of a lock-in amplifier which measures the transmission of light through the sample. The data shown represent an unnormalized transmission spectrum for a GaAs sample.

an indirect band gap semiconductor (specifically, GaP) is studied. The benefits of the automation provided by LabVIEW are generally quickly appreciated by the students.

IV. LabVIEW AND COMPUTER-AIDED LEARNING

We have been gratified to find from both formal and informal feedback that the majority of those students who find conventional programming difficult are significantly more comfortable with creating LabVIEW code. Those with a strong background in computer programming using high level languages such as C++ or Java take somewhat more time to ‘‘warm’’ to LabVIEW. However, for both sets of students a particular benefit (cited in student questionnaires) of the course is the reinforcement and strengthening of concepts related to Fourier analysis.

From an instructor’s point of view, it is clear that the ability to rapidly plot the real and imaginary parts of Fourier transforms of simple (and not so simple) waveforms is an important pedagogical tool. In particular, it enables the development of an ‘‘intuition’’ related to the relationship between time and frequency space (or direct and reciprocal space) that is unfortunately sometimes not easily derived from purely analytical mathematics courses. For example, we find that it is not rare for a student to fail to appreciate that the more time-limited a function/waveform, the greater its associated bandwidth, despite that student having analytically calculated very many Fourier integrals.

In addition to strengthening concepts covered in theoretical lecture modules, a very important element of the LabVIEW course described here is its problem-solving character. A criticism that often appears in student questionnaires related to the course is that the laboratory/course manual seemingly does not provide enough detail with regard to completing the assigned tasks and projects. This perceived lack of detail is deliberate; considerable effort has been expended in ensuring that the course manual is not overly prescriptive. Where possible the students are expected to develop their own solutions to problems and, most importantly, question the data that any LabVIEW code they develop provides. Similarly, the low level of assessment in the early part of the course (work on only two of the first six lab sessions is assessed) is designed to enable students to work at their own pace and gain a relatively deep understanding of the material.

In addition to the second year course described in this paper, LabVIEW is used for project work in the third and fourth years of Nottingham physics degree courses. In our experience (based on the use of LabVIEW throughout our degree courses), there are a number of ‘‘perennial’’ issues:

(i) the absence of error bars on LabVIEW graphs;
(ii) the importance of ensuring that students understand the concept of dataflow and the use of various ‘‘tricks’’ (such as the wiring of error inputs/outputs) to determine the order of execution of the various nodes and sub VIs in a block diagram;
(iii) matrix/array manipulation requires careful explanation (in particular, the means by which arrays are transposed for data display/storage can cause confusion);
(iv) the concept of polymorphism can be somewhat difficult for the majority of students to grasp (in our case, only a ‘‘vanishingly small’’ minority of the students taking the course are familiar with object oriented programming); and
(v) introduction of the *waveform* data type in LabVIEW 6.1 has led to complications in explaining the use of clusters as data structures in *G*.

Although these issues may at times be frustrating, they are hardly insurmountable and do not affect the fundamental premise of this paper: LabVIEW is an extremely powerful resource for undergraduate (and postgraduate) teaching and training.

V. CONCLUSIONS

LabVIEW is a novel and powerful platform on which to base a modern undergraduate laboratory course in the physical sciences. In addition to providing students with training in interfacing and digital signal processing techniques, the course we have described has particular pedagogical advantages in teaching concepts such as Fourier analysis which are at the core of a physics undergraduate’s education.

ACKNOWLEDGMENTS

We thank our colleagues in Physics and Astronomy in Nottingham who have provided helpful comments and criticisms regarding the LabVIEW course. These include Alfonso Aragon-Salamanca, Steve Maddox, David Rourke, Peter Beton, Reynier Peletier, and Paul Glover. Feedback from graduate demonstrators (in particular Mick Phillips, Robin Scott, and David Sherwood) has also been beneficial in improving elements of the course. Bill Moseley, the second year laboratory technician, provided invaluable assistance in the day-to-day running of the lab. We also very gratefully acknowledge the work of both Paul Reynolds and Phil Hawker in maintaining and upgrading the computing facilities. Finally, we would like to thank all the undergraduate students who have provided very helpful feedback on the course over the past four years.

[1] Lisa Wells and Jeffrey Travis, *LabVIEW for Everyone: Graphical Programming Made Even Easier* (Prentice Hall, Englewood Cliffs, NJ, 1997).

[2] John Essick, *Advanced LabVIEW Labs* (Prentice Hall, Englewood Cliffs, NJ, 1999).

[3] P. Moriarty, *Interfacing and Modern Measurement Techniques*, 2nd year Laboratory manual, School of Physics & Astronomy, University of Nottingham (1998).

[4] M. L. Chugani, A. R. Samant, and M. Cerna, *LabVIEW Signal Processing* (Prentice Hall, Englewood Cliffs, NJ, 1998).

[5] K. Turvey, ‘‘An undergraduate experiment on the vibration of a cantilever and its application to the determination of Young’s modulus,’’ Am. J. Phys. **58**, 483–487 (1990).

[6] W. J. Parker, R. J. Jenkins, C. P. Butler, and G. L. Abbott, ‘‘Determination of thermal diffusivity using a flash method,’’ J. Appl. Phys. **32**, 1679–1686 (1961).

[7] S. J. Rogers, in *Physics Experiments and Projects for Students, Vol. I*, edited by C. Isenberg and S. Chomet (The Institute of Physics, UK, 1985).

[8] R. L. Zimmerman, S. Celaschi, and L. G. Neto, ‘‘The electronic bouncing ball,’’ Am. J. Phys. **60**, 370–375 (1992).

[9] J. M. Essick and R. T. Mather, ‘‘Characterization of a bulk semiconductor’s band gap via a near-absorption edge optical transmission experiment,’’ Am. J. Phys. **61**, 646–649 (1993).

[10] L. Martil and G. Gonzalez-Diaz, ‘‘Undergraduate laboratory experiment: Measurement of the complex refractive index and the band gap of a thin film semiconductor,’’ Am. J. Phys. **60**, 83–86 (1992).