INTRODUCTION TO LABVIEW TWO-PART EXERCISE

Sami D. Alsaialy\textsuperscript{1}, Dalia M. Tawy\textsuperscript{2}, and Susan M. Lord\textsuperscript{3}

Abstract - At the University of San Diego, we wanted to introduce all electrical engineering (EE) students to LabVIEW without adding additional courses. In Fall 2001, juniors in Electronics performed an “Introduction to LabVIEW” exercise designed by two EE juniors. Students performing the exercise obtained the amplitude frequency response of an RC circuit using a manually controlled function generator and a computer-controlled oscilloscope. During Summer 2002, two EE seniors improved the LabVIEW program to include measurement of phase, work with a computer-controlled function generator, and have more readable code. In Fall 2002, juniors in Electronics performed a two-week introduction to LabVIEW. The first week was the same as in Fall 2001. In the second week, the two programmers delivered a presentation focusing on the benefits of LabVIEW, a comparison of LabVIEW and other programming languages, and a discussion and demonstration of their enhanced program. Finally, the juniors wrote a simple LabVIEW program.

Index Terms - circuits, electronics, frequency response, LabVIEW, student-designed laboratories.

INTRODUCTION

National Instruments’ LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that is commonly used throughout academia, industry, and government. Thus some familiarity with this vital tool is important for undergraduate electrical engineering (EE) students. Given the wealth of topics in modern engineering education, it is challenging to find a place in the curriculum to introduce LabVIEW. At the University of San Diego (USD) in Fall 2002, we chose to introduce LabVIEW during the first two weeks of the junior level Electronics I class. This introduction was also used as a vehicle for two-tiered learning whereby senior EE students developed the introduction for junior EEs. Both groups of students benefited from the experience.

For this “Introduction to LabVIEW”, we chose to focus on frequency response measurements which are commonly used to characterize the behavior of electrical circuits and junior EEs are already familiar with them from their sophomore circuits’ class. Since the first laboratory in Electronics I met after only one day of lecture, students are often frustrated in trying to complete labs on new Electronics topics before covering material in class. This “Introduction to LabVIEW” was designed to be easy for the students to follow and did not introduce any new laboratory measurements. Rather it allowed students to consider the advantages of automating a measurement. As juniors, they will perform frequency response measurements many times throughout their two-semester Electronics sequence (EE 130 and 132). Although it is important for them to perform the measurements manually the first few times to be sure they understand what they are doing, the ability to automate such measurements allows students to focus on design and analysis later in their study of analog circuits. LabVIEW is an excellent candidate for such automation. We introduced the juniors to automation using LabVIEW gradually. During the first week of the “Introduction to LabVIEW”, they performed the measurements with a manually controlled function generator and a simple LabVIEW program to plot the amplitude-frequency response of an RC (Resistor – Capacitor) filter. [1] In the second week, they saw how the automation would be improved with a GPIB (General Purpose Interface Bus)-controlled function generator and an enhanced LabVIEW program developed by two EE seniors (authors SDA and DMT) under the supervision of one of the authors (SML) at USD during Summer 2002. The two senior students who designed the new program gave a presentation to the juniors about the advantages of LabVIEW and the new LabVIEW frequency response program. A demonstration of the new program was given and then the juniors created a simple LabVIEW program based on a handout written by the two senior students.

LABORATORY EXERCISE: WEEK 1

During Fall 2002, all junior EEs in Electronics I performed the first “Introduction to LabVIEW exercise” in groups of two with each taking about 1 hour. The junior students were supplied with a simple LabVIEW user’s manual written by a senior USD student.[2] The students ran a LabVIEW program that plots the amplitude frequency response of the simple RC circuit shown in Figure 1. The program controlled an oscilloscope automatically while the students created a frequency sweep manually using a function generator. The students also added a small part to the LabVIEW program to calculate the cut-off frequency of the circuit theoretically by following a step-by-step handout.[3] After collecting data, the students used the cursors on the frequency response graph to estimate the experimental value of the cut-off frequency. The students compared the
theoretical and the experimental values of the cut-off frequency. The students wrote a one to two paragraph summary in which they explained what they learned from the lab, and suggested possible improvements. The front panel of the LabVIEW program used for this exercise is shown in Figure 2. The equipment used for the experiment consisted of an HP 54600A oscilloscope with GPIB, a Wavetek 148 Function Generator, and a personal computer with LabVIEW 6i. Multiple stations were provided.

In the second week, the lab consisted of two major sections: a PowerPoint presentation and an exercise. At the beginning of the lab period, two senior students (SDA and DMT) gave a presentation including an introduction to LabVIEW and its uses, as well as an explanation of the advantages and disadvantages of LabVIEW. The major advantages discussed were the ease of learning, using and debugging graphical programming languages such as LabVIEW, the simplicity of using the interface (front panel of a LabVIEW program) especially for a user with a little or no knowledge of LabVIEW programming, the capability of controlling equipment using GPIB, and the availability of the LabVIEW drivers for most measurement equipment on line at the National Instruments website (http://www.ni.com) and on the library CD that accompanies the LabVIEW software. The disadvantages discussed included the high cost of the LabVIEW software, its upgrades and the accompanying hardware (GPIB board or GPIB-USB cable), and the occasional crashing of the programs. The presentation also demonstrated a comparison between LabVIEW and other programming languages such as C++ which all USD engineering students study in their sophomore year. The comparison included showing the code of a simple addition program using C++ and LabVIEW as shown in Figure 4 and Figure 5, respectively. The students could clearly see the difference between a graphical programming language (LabVIEW) and a syntax programming languages (C++).

In the second part of the presentation, the presenters elicited comments from the students about the problems they encountered with the frequency response LabVIEW program they used the previous week. The students mentioned most of the following problems:

1. The program only controls the oscilloscope automatically, while the user has to control the function generator manually by setting the increments of the frequency.
2. The program only works for the specific RC circuit being used because all of the settings of the program are based on this circuit. If the user wants to plot the amplitude frequency response of another circuit, he or she has to determine the initial settings before using the program.
3. The user has to change the time per division of the oscilloscope manually using the front panel of the program while the program is running in order to be able to gather data points at higher frequencies than the initial frequencies.
4. The user has to specify the Excel sheet extension to save the data to Excel.
5. There are some incorrect points collected and graphed by the program.
6. The user has to end the program by shutting the program manually.
7. The program only plots the amplitude frequency response of the tested circuit. Plotting the phase versus frequency of the circuit provides for complete frequency response data.
8. The code of the program is complicated and confusing. There are not enough labels and documentation.
Figure 3
Front panel of new LabView program.

Figure 4
The code of a C++ program for calculating cutoff frequency:

```c++
#include <iostream>

void main()
{
    #define R 1000000
    #define C 1000
    #define pi 3.1415
    float R = (float)R;
    float C = (float)C;
    float pi = (float)pi;
    float f = (R*C)/pi;
    std::cout << "The cutoff frequency is: " << f << std::endl;
}
```

Figure 5
The code and front panel of a LabView program for calculating cutoff frequency.
In the third part of the presentation, the presenters explained the following advantages of the new LabVIEW program shown in Figure 3 highlighting how it solves the problems encountered in the old program:

1. The program controls the function generator automatically through the GPIB connection. (Note that this requires a GPIB interface to be available on the function generator.) The user specifies the values of the initial frequency, the final frequency and the frequency increments on the front panel before he or she runs the program as shown in Figure 3. A frequency sweep is generated according to the following equation:

   \[ \text{New Frequency} = \text{Previous Frequency} + (\text{Number of the order of the point} \times \text{increment value}) \]

   so the user does not have to increment the frequency manually.

2. The program can measure a variety of electrical circuits such as high-pass filters, low-pass filters, and bandpass filters or transistor amplifiers. The oscilloscope settings (e.g. time/div, voltage/div) are adjusted automatically according to the circuit being tested.

3. The time per division setting is adjusted automatically as each frequency data point is measured according to the following equation:

   \[ \text{Time/Division} = \frac{1}{2 \times \text{Frequency}} \]

   After running the program, the two presenters explained the major components of the LabVIEW code used in the program. The presentation ended with a question session.
After the presentation and demonstration, the junior students wrote a simple LabVIEW program by following step-by-step instructions [4]. Based on an exercise in an introductory LabVIEW book [5], the program calculates and plots the velocity and acceleration of any provided displacement equation. The students had to write the code and investigating the graphs on the front panel. The front panel and the diagram window of the displacement-velocity-acceleration program are shown in Figure 9 and Figure 10, respectively. For their write-up, the students submitted a summary explaining what they learned and suggesting possible improvements for the two-week introduction to LabVIEW.

**RESULTS AND STUDENT FEEDBACK**

This project provided an excellent opportunity for two-tiered learning. The two senior EE students learned about programming in LabVIEW, designing laboratories for their colleagues, preparing handouts and presentations, and supervising students in the lab. One of the students (DMT) had previous experience with LabVIEW during a Summer 2001 research experience at USD. The other student (SDA) started learning LabVIEW at the beginning of Summer 2002. By the end of this project, the two students gained a lot of experience with LabVIEW programming which enhanced their knowledge and resumes. The students also improved their communication and documentation skills through preparing and presenting the PowerPoint presentation as well as guiding the juniors who performed the lab. Getting feedback from the juniors was also valuable for helping the senior students to improve their performance in future presentations and technical communication.

None of the juniors who performed the two-week laboratory were familiar with LabVIEW before this introduction. Thus they all learned about topics such as LabVIEW, its capabilities, the value of automating a
commonly used measurement, and graphical programming languages. Most of the juniors showed an appreciation for LabVIEW and interest in learning more about it. All of them were able to complete the final exercise in which they used the programming language individually and many commented that they enjoyed this. Most of them also found the presentation beneficial.

Here are some of the actual comments of the students:

- The main advantage to LabVIEW is the fact that it is a graphical programming language. A language coded in the manner of C++ or Matlab has no graphical diagram interface similar to what LabVIEW. Data collection in LabVIEW takes place instantaneously. Collecting a simple oscilloscope screen capture can take a few minutes by trying to import it Excel, and sometimes it doesn't work at all.
- I felt that the strongest part of the demonstration was that the 2 presenters had a very strong understanding of the material. They explained things very concisely and clearly which made a big difference in helping to understand the program. It also helped seeing a demonstration done because it shows you what it should look like when you are finished, so if ours looked different from theirs, then we would know that we went wrong somewhere. The actual exercise was the best part about the lab because the directions that were given were very concrete and easy to follow.
- Possible Improvements: Shorten presentation time and let students make two programs, however the presentation was very good.

The two-week format allowed for the gradual introduction of automation with LabVIEW. Equipment constraints prohibited us from having multiple stations with GPIB-controlled function generators and thus using the new program during the first week. Multiple stations are necessary for all the students to be able to perform the lab. The exercises developed here could be adapted to different formats. For example, one alternative might be to run the lab over two weeks but divide the class into two groups. Thus during the first week, one group could experiment with the new program and several LabVIEW programming exercises and the second group could do these activities the next week. The presentation could be adapted to stress the advantages of LabVIEW and minimize the comparisons between the old and new program depending on time constraints. In its current form, a nice feature of the presentation was the emphasis on problem solving where juniors had an opportunity to identify problems which the seniors had previously identified and solved. However the introduction is done, it is important that students understand the measurements they are making and not just think of LabVIEW as a black box that produces data. An additional benefit of this project is that the presentation was also done for other EE faculty members and helped convince them to buy more GPIB-controlled function generators so LabVIEW could be used throughout the Electronics sequence in future years.

**SUMMARY**

In Fall 2002, sixteen electrical engineering juniors at USD performed a two-week laboratory in their Electronics I course which introduced them to LabVIEW. In the first week, the students plotted the amplitude frequency response of an RC circuit using LabVIEW, a GPIB-controlled oscilloscope, and a manually controlled function generator. In the second week, the same students attended a presentation highlighting the uses and capabilities of LabVIEW given by two senior USD EE students. The seniors also demonstrated an enhanced LabVIEW program which performed amplitude and phase frequency response using a GPIB-controlled oscilloscope and function generator. Finally, the junior students individually wrote a simple LabVIEW program to plot the velocity and acceleration of a displacement equation. Both the senior and junior students benefited from the experience. The seniors gained technical experience with LabVIEW as well as experience in written and oral technical communication. Overall, the junior students enjoyed the two-week experience, learned enough about LabVIEW to write a simple program, and showed interest in learning more about LabVIEW.

**ACKNOWLEDGMENT**

The authors acknowledge the suggestions, feedback, and cooperation of the students in the Electronics class (EEE 130) at USD in Fall 2002.

**REFERENCES**


**SESSION T4E**