AC 2009-1334: ON THE USE OF LABVIEW IN SIGNALS AND SYSTEMS

Jayaraman Jayaraman Thiagarajan, Arizona State University

Kostas Tsakalis, Arizona State University

Andreas Spanias, Arizona State University

Harvey Thornburg, Arizona State University

ON THE USE OF LABVIEW IN SIGNALS AND SYSTEMS

1. Introduction

Computer based data acquisition and instrumentation control packages are embedded in several industrial and education applications. The National Instruments **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench (LabVIEW) package is tailored for data acquisition, data processing and instrumentation control. LabVIEW is a graphical programming language and can interface with external acquisition and signal processing devices[1].

In this paper, we describe the use of LabVIEW in undergraduate signals and systems courses. The understanding of signals and systems is central to several areas in Electrical and Computer Engineering. Signals and systems courses serve as prerequisites for higher level courses in signal processing, communications and controls[2,3]. The availability of a wide range of functions and toolkits for in LabVIEW enables exposition to algorithm, software and hardware issues in signal analysis and filtering. We have started an education project with National Instruments aimed at developing and embedding software and laboratory exercises for signals and systems courses. The main objectives of the exercises are to provide students with an introduction to graphical programming and also enable them simulate the concepts learnt in the course. These labs provide hands-on experiences associated with concepts in signals and systems, DSP, communications, and controls. Students first learn the fundamentals of LabVIEW graphical programming including array and cluster processing, loops and file handling. Furthermore, interesting features of LabVIEW such as the Mathscript functionality and simulation loops are introduced. Students are also exposed to different toolkits such as Digital Filter Design, Control Design and Modulation. The rest of the paper is organized as follows. Section 2 introduces the LabVIEW functions and toolkits used in the laboratory exercises. The signals and systems LabVIEW exercises are given in Section 3. Finally, the assessment of the labs and student responses are presented in Section 4 and the paper is concluded in Section 5.

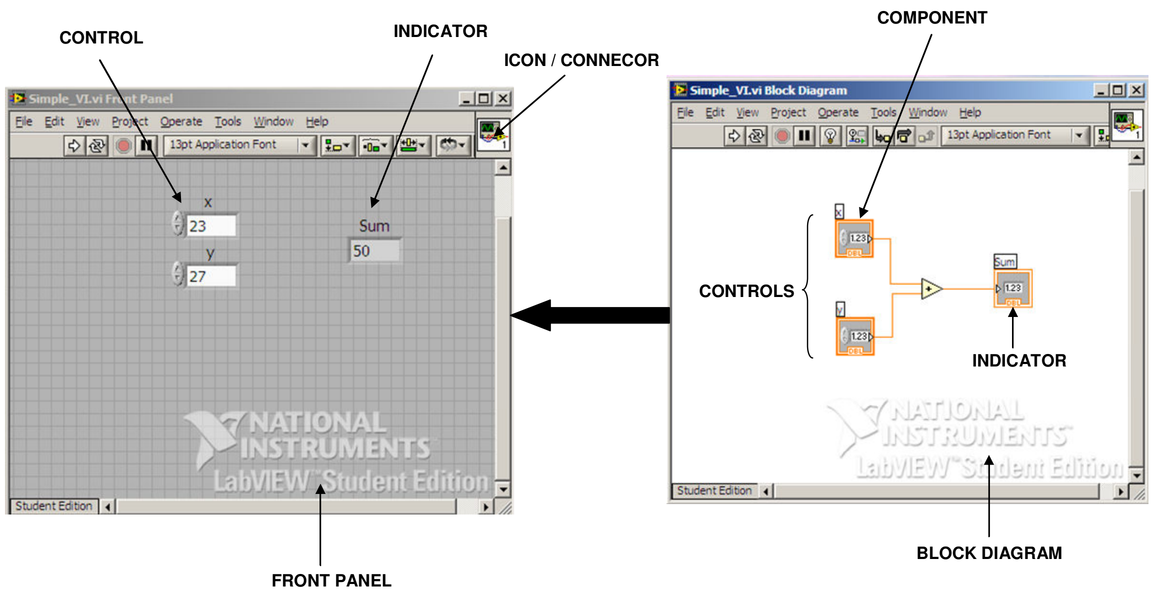

Figure 1. Illustration of components of a VI.

2. LabVIEW in Signals and Systems

LabVIEW enables users to develop applications that are interactive, executed in parallel and multicore. LabVIEW programs are called Virtual Instruments (VIs). Each VI has three main parts: the block diagram, the front panel and the icon/connector. The block diagram contains the actual graphical code, while the front panel is the user interface. Controls and indicators on the front panel allow the user to input data into or extract data from a running virtual environment. Each VI in turn can contain sub VIs and other structures. This hierarchical programming structure allows code reusability and provides modularity to the program. Figure 1 illustrates the components of a VI.

LabVIEW has a rich palette of functions for signal analysis and system control, that enables students to simulate and visualize concepts easily[4,5]. In this section, we describe some of the LabVIEW functionalities that are used by the undergraduate students to perform their laboratory exercises.

2.1. Express VIs

Express VIs are LabVIEW functions used to build common measurement tasks and they enable the user to interact graphically to modify the parameters. The input Express VIs can be used to acquire or simulate signals, while the set of output Express VIs are used to save data to files, communicate with instruments, and display messages to users. Furthermore, LabVIEW contains a rich set of Express VIs for signal analysis and signal manipulation. Figure 2 illustrates some of the Express VIs commonly used by students in the labs.

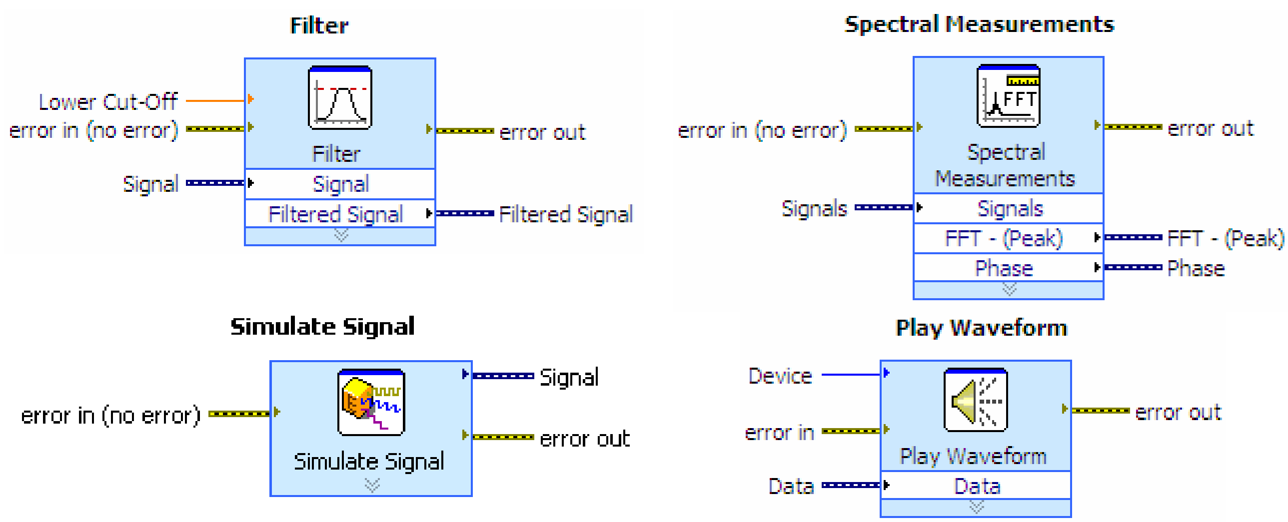

Figure 2. Express VIs commonly used in the exercises.

2.2. Simulation loop

The response of a linear system to arbitrary inputs can be computed using *simulation VIs* in LabVIEW. One can create a simulation diagram by placing all the simulation functions and subsystems inside the *simulation loop*. Simulation loop executes the simulation diagram until the *Final Time* is reached or the simulation is halted programmatically. Figure 3 illustrates a simulation loop in LabVIEW.

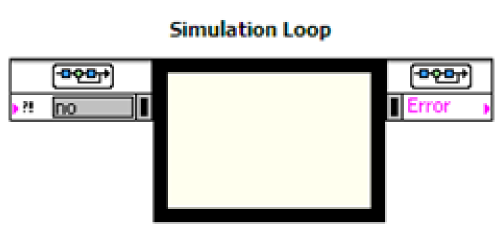

Figure 3. Illustration of a simulation loop.

2.3. Mathscript

LabVIEW graphical programming paradigm can be used with LabVIEW MathScript, a math-oriented textual programming language that is generally compatible with the widely used *.m* file. One can type the MathScript commands inside the blue structure or import a MATLAB

.m file directly. Mathscript can be used to communicate with other LabVIEW functions. Figure 4 illustrates the use of a Mathscript node.

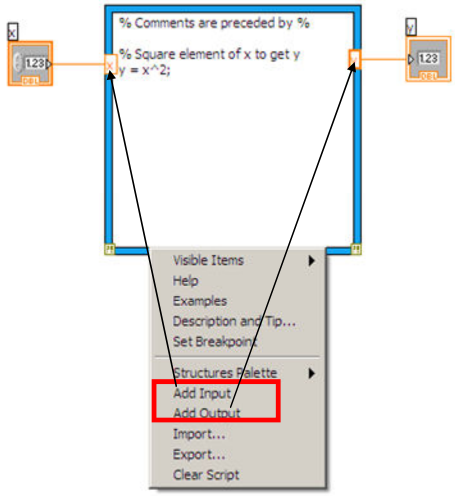

Figure 4. Illustration of a Mathscript node.

2.4. Digital Filter Design Toolkit

The LabVIEW Digital Filter Design (DFD) toolkit includes several advanced filter design tools for designing, analyzing, and simulating floating-point and fixed-point digital filters[6]. The DFD toolkit includes Express VIs, which can be used to interact graphically with filter specifications to design appropriate digital filters. In addition to tools that help the students create digital filters, the DFD toolkit includes tools for conventional and multi-rate digital filter design, floating-point to fixed-point conversion and filter analysis. Figure 5 shows some of the toolkit functions used by the students to design digital filters.

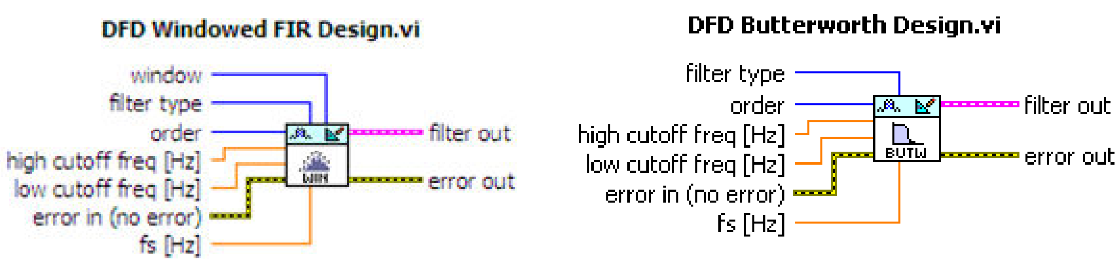

Figure 5. DFD toolkit functions used by the students.

2.5. Control Design Toolkit

The Control Design (CD) toolkit provides a library of VIs and LabVIEW MathScript functions that can be used to design, analyze, and deploy a controller for a linear time-invariant dynamic system model[7]. This toolkit includes frequency response analysis tools such as Bode plot, time response analysis tools such as step and impulse response analysis and classical design tools such as Root Locus. Figure 6 shows some of the toolkit functions used by the students to design and analyze an LTI system model.

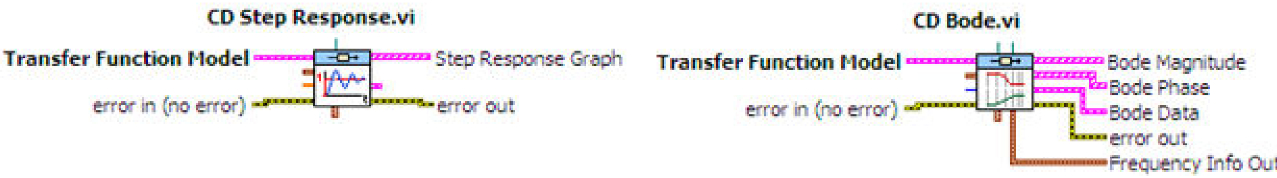

Figure 6. CD toolkit functions used by the students.

3. Laboratory Exercises

The students are initially provided with an introduction to graphical programming using LabVIEW and they are introduced to some of its signal analysis related features. The first exercise involves building of a VI to load and display real-time speech data frame-by-frame. In addition, they will filter the input speech using a low pass filter, whose cut-off frequency can be dynamically varied. The signal length (in samples) and the frame count (depends on the frame size) will also displayed in the Front Panel. An option to playback the filtered speech will also be provided. This exercise emphasizes the use of Express VIs, basic array processing, simple filtering and creation of sub VIs. This provides students with the necessary background in LabVIEW to perform more advanced processing. The exercises that the students performed in the course are listed in Table 1. In this section, we describe two sample exercises that the students performed as a part of their course curriculum.

Basic speech processing

- Time-domain and frequency domain analysis of an input speech file.

Filter design and analysis

- Analysis of a continuous-time LTI system.
- Design of digital FIR and IIR filters.

Sampling, aliasing and equalization

- Study of the aliasing effect.
- Analysis of upsampling.
- Design of a music equalizer.

Amplitude modulation and demodulation

• Time-domain and frequency-domain analysis of the modulation effects.
Linear feedback
• Demonstration of a PI controller.

Table 1. List of LabVIEW exercises performed by the students.

3.1. Analysis of a continuous-time LTI system

In this exercise, the students create and analyze a continuous-time LTI system and convert it into a discrete-time system. Furthermore, they evaluate the response of the designed LTI system to arbitrary signals. A transfer function of an LTI system, given by the Laplace transform of its impulse response, is a transform-based mathematical representation of the relation between the input and output of the system. Using the functions of the CD toolkit, the students create an LTI system using the transfer function model and compute the Bode response of the designed system.

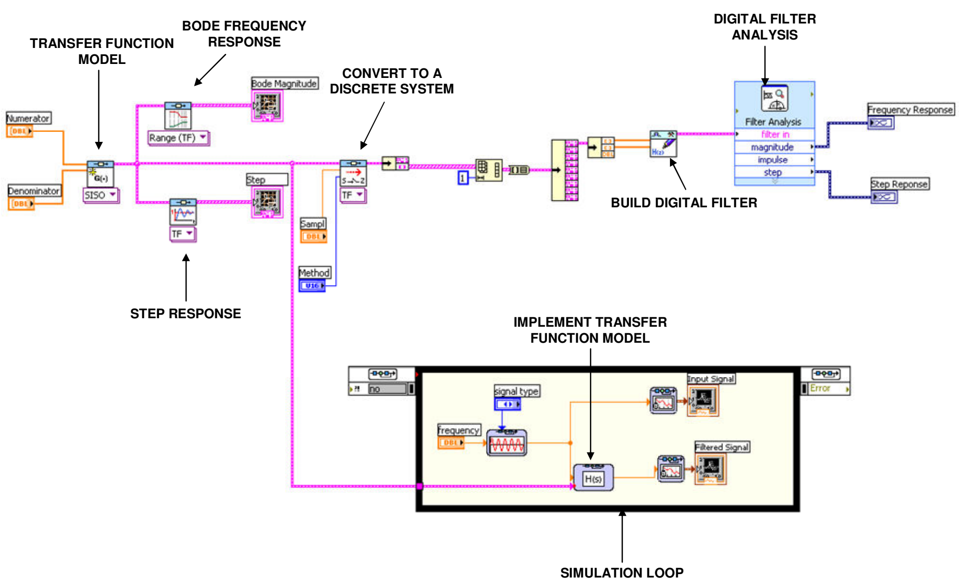

Figure 7. Graphical code for the analysis of a continuous-time LTI system.

The continuous transfer function model is converted to a discrete system using the zero-order hold method and the step response of the LTI system is compared to that of its discretization. The response of the linear system to arbitrary inputs is determined using *simulation loops* in LabVIEW. Figure 7 and Figure 8 show the block diagram code and the front panel GUI of this exercise.

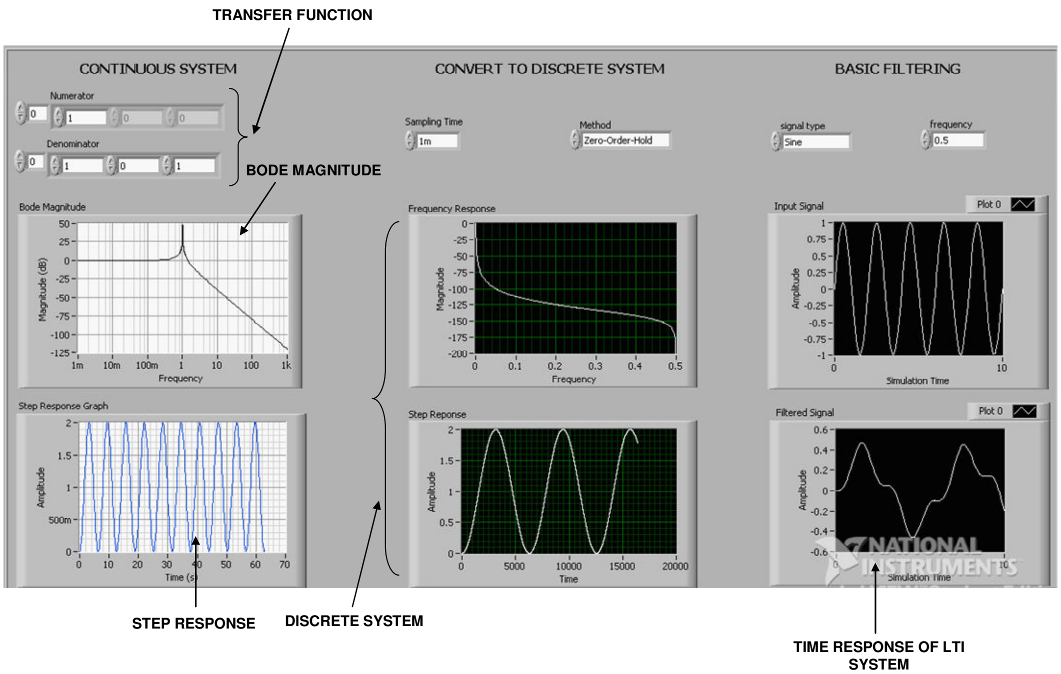

Figure 8. Front panel for the analysis of a continuous-time LTI system.

3.2. Design of a music equalizer

Equalization is the process of altering the frequency response characteristics of a signal. The graphic equalizer is very common in music applications.

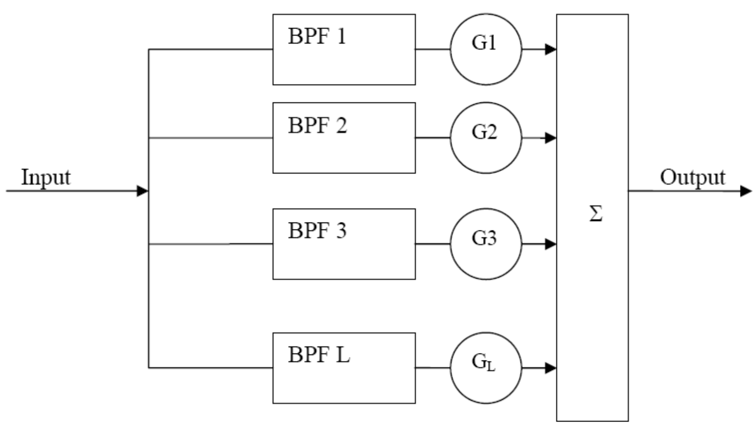

Figure 9. Design of a simple equalizer.

A music equalizer is used to amplify or attenuate a particular band of frequencies of a given signal in order to get better sound effects. Figure 9 shows the design of a basic equalizer. As shown in the figure, the input signal is divided into different frequency bands by a series of bandpass filters (BPF) and then each band is attenuated or boosted by different Gain factors (G) and all the bands are added finally to generate a composite signal. The frequency bands to be used in the design are provided to the student.

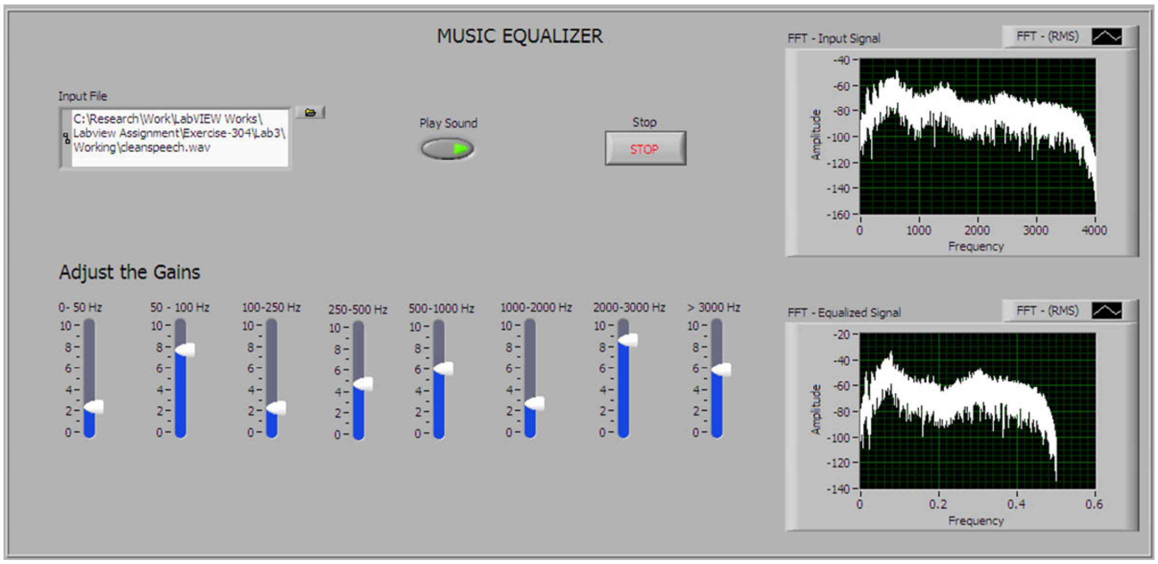

Figure 10. Front Panel Layout for a music equalizer.

4. Assessment

The LabVIEW labs were performed by the undergraduate students of EEE 304 at Arizona State University. We performed summative assessments based on student feedback. In Spring and Fall 2008, a total of 79 students participated in the evaluation and the overall response was found to be positive. The questions posed in the assessment quiz and the student responses are itemized in Table 2. The results indicate that the students gained significant hands-on experiences with the LabVIEW exercises and they felt comfortable in using LabVIEW for simulating some important concepts in signals and systems.

The primary advantage that the students noted was the ease of implementation of algorithms and concepts. They also commented positively on the graphical interface. They felt that the experiments on aliasing and upsampling real-time signals with the aid of LabVIEW functions reinforced the fundamental concepts of sampling. Furthermore, the students have reported that the exercises on the design of a music equalizer and the amplitude modulator/demodulator were very educational. Details on the questions posed and responses obtained are given in Table 2.

Question	Strongly agree	Agree	Disagree	Strongly disagree
By performing the LabVIEW exercises, you became familiar with the graphical programming practices.	30	49	0	0
The basic graphical programming information provided in the first exercise was sufficient to perform the tasks.	23	53	3	0
LabVIEW made learning signal processing concepts easier and intuitive.	24	52	3	0
You now feel comfortable in choosing the LabVIEW functions to design a block diagram.	70	0	9	0
The contents of the exercise in Lab 1 helped you understand some basic concepts and tools used in speech processing.	16	60	3	0
The exercise on Aliasing (Lab 3) helped you understand the effects of oversampling	75	0	4	0
Performing the exercises in Lab 2, helped in understanding the basics of filter design using LabVIEW.	21	52	6	0
Array handling in LabVIEW was made clear through the exercises.	44	0	35	0
Performing Lab 2, helped you learn to use the LabVIEW Control Design Toolkit and the Digital Filter Design Toolkit.	67	0	12	0
You are now comfortable in creating subVIs and using them in your block diagram.	20	56	3	0

Table 2. Results of the assessment administered to the undergraduate students.

5. Conclusions

In this paper, we presented the development of several laboratory signals and systems exercises based on the National Instruments LabVIEW. More specifically we developed introductory computer laboratories on sampling, filtering, modulation, and controls. Students learned how to program and execute simple signals and systems software modules using LabVIEW. An assessment of these exercises was administered in the Spring and Fall 2008 semesters. The student responses indicate that they found the use of software to experiment with theoretical concepts in signals and systems quite useful. They also noted the ease of programming with LabVIEW. Specifically, 89% of the students reported that they became familiar enough with LabVIEW and its use in signals and systems and they can develop their own LabVIEW programs. Student responses reveal that the visualization tools helped students improve understanding of fundamental concepts such as aliasing and upsampling (95% of students). In addition to the basic LabVIEW functions, the students responded that the functions provided in the Digital Filter Design and Control Design toolkits enabled them to design simpler system representations.

Bibliography

1. Beyon, J. Y., "LabVIEW Programming, Data Acquisition and Analysis", Prentice Hall PTR (Upper Saddle River, NJ), 2001.

2. Oppenheim, A.V., Willsky, A.S. and Hamid, S., "Signals & Systems," Prentice Hall Signal Processing Series, Hall PTR, (Upper Saddle River, NJ).

3. Spanias, A., "Digital Signal Processing; An Interactive Approach, Textbook with JAVA-DSP computer exercises", ISBN 978-1-4243-2524-5, Tempe, February 2007.

4. Wells, L.K. and T. Jeffrey, "LabVIEW for Everyone: Graphical Programming Made Even Easier", Prentice Hall PTR, (Upper Saddle River, NJ), 1996.

5. Jamal, J. and H. Pichlik, "LabVIEW Applications and Solution", Prentice Hall, (Upper Saddle River NJ), 1999.

6. Digital Filter Design toolkit manual, Available at http://sine.ni.com/manuals/.

7. Control Design toolkit manual, Available at http://sine.ni.com/manuals/.