A First Course in the Design of Data Acquisition Systems

ROBERT T. DOTY

Department of Engineering
Baylor University

Abstract

The computer revolution has created exciting new opportunities for automating engineering laboratories. With standard off-the-shelf computers and low-cost data acquisition systems, students can acquire and analyze data from sophisticated laboratory experiments that were previously impractical. This paper describes an introductory course at Baylor University in the design of these computer-assisted data acquisition systems. The designs are implemented as virtual instruments using National Instruments LabVIEW™ graphical programming language. The goal of this course is to provide knowledge and experience in the design and implementation of computer-assisted data acquisition and computer-controlled instrumentation and is open to all engineering students who have been admitted to the engineering upper division.

The course emphasizes "learning by doing" through a series of hands-on learning modules. One of these learning modules is discussed in some detail in order to demonstrate this educational approach. This module provides a real-world digital signal processing example by decoding a seven-digit touch-tone telephone number. The student acquires 1.4 seconds of touch-tone data; decodes, displays, and plays the telephone number through the computer's sound card; and identifies the caller from a database. This module illustrates how information can be extracted from a dual-tone multiple frequency signal using bandpass digital filters.

Introduction

This paper describes an introductory course at Baylor University in the design of computer-assisted data acquisition systems. The designs are implemented as virtual instruments using National Instruments LabVIEW™ graphical programming language. The goal of this course is to provide knowledge and experience in the design and implementation of computer-assisted data acquisition and computer-controlled instrumentation and is open to all engineering students who have been admitted to the engineering upper division.

The course emphasizes "learning by doing" through a series of hands-on learning modules. The course is structured such that a two-week project module follows several one-week homework modules. The homework modules are designed to provide both the LabVIEW skills and the engineering knowledge necessary to design and implement data acquisition systems. Students are allowed to consult with one another and seek help from the course instructor when working on a homework module. The project modules are designed to integrate the material just covered in the preceding homework modules and are to be implemented without outside consultation, thereby serving as take-home exams.

The instructor evaluates the student's performance by executing the submitted LabVIEW source code, which the student copies to the instructor's secure drop box on the engineering fileserver at submittal time.

It should be mentioned that the student edition of LabVIEW is a recommended purchase for the course. The book that accompanies the student edition is not used as a text but serves as a valuable supplemental student resource.[1] Many engineering students own their own computer and prefer to work at home. The student edition is very robust and is fully compatible with the professional version installed in the engineering computer laboratories.

Another important feature introduced in this course is remote data acquisition and remote instrument control over local area networks and the Internet. The National Instruments DataSocket™ Server and protocol is utilized as a simple and reliable means for publishing and subscribing to remotely acquired data.[2]

Additional course details may be found at
http://ecswww.baylor.edu/faculty/doty/EGR3310/EGR3310.html

The Learning Modules

The course topics by week are as follows:
(1) Introduction to LabVIEW (HW1)
(2) Acquiring and Storing Data (HW2)
(3) Retrieving and Analyzing Data (HW3)
(4) Digital and Analog I/O (HW4)
(5) Design of a Basic DAQ System (PR1)
(6) Submittal of Project 1
(7) Fourier Series and FFT (HW5)
(8) Digital Filtering (HW6)
(9) Spectrum Analysis & Aliasing (HW7)
(10) Design of a DSP System (PR2)
(11) Submittal of Project 2
(12) Oversampling and Averaging (HW8)
(13) Damper Motor Calibration (HW9)
(14) Dead Band Controller (HW10)
(15) Design of a Control System (PR3)

A brief decription of each of the learning modules is presented below. Notice that he notion of toggling between virtual and laboratory data acquisition is introduced early in the course and used throughout the semester. This allows the student to development and test virtual instruments on a computer that has no DAQ card.

HW1: Introduction to LabVIEW

Scope

Implement a LabVIEW virtual instrument that generates 1001 points from any one of several standard waveform types and plots them on a waveform graph. Convert the x-axis into a one second time display. Continuously run the instrument using a WHILE Loop with a Push to Stop button and interactively select the waveform type from a front panel control.

Purpose

This exercise will introduce you to the LabVIEW programming environment and illustrate how the front panel, block diagram, and icon/connector work together to produce a virtual instrument that can stand alone or be called by another program.

HW2: Acquiring and Storing Data

Scope

Implement a LabVIEW virtual instrument that acquires a timed analog waveform and writes the data to a spreadsheet file. Use a switch to toggle between virtual data and laboratory data. Read the file into Excel and graph the data.

Purpose

This exercise will introduce you to the while, case and sequence structures, basic analog data acquisition, array manipulation, analysis of data, charting and graphing, and file handling. In addition, you will share the acquired data with another application.

HW3: Retrieving and Analyzing Data

Scope

Implement a LabVIEW virtual instrument to be used to read in an underdamped sinusoidal waveform from a text file and analytically determine its decay constant.

Purpose

This exercise will introduce you to data retrieval from a spreadsheet file, regression analysis using the Curve Fit.vi and illustrate one method of determining the decay constant from experimental observation.

HW4: Remote Data Acquisition

Scope

Design and implement a LabVIEW virtual instrument that defines the frequency of an audible pure tone and publishes it for remote processing.

Purpose

This exercise will introduce you to remote data acquisition, wave files, sound card control, and the creation of a *vi* library.

PR1: Design of a Basic DAQ I/O System

Scope

Design and implement a LabVIEW virtual instrument that acquires lab data and addresses a lab instrument. Demonstrate both digital I/O and analog I/O. Use a switch to toggle between virtual data and laboratory data. System performance shall be verified through interaction with the IDL-800 Digital Lab at the compliance test.

HW5: Fourier Series and FFT
Scope
Implement a LabVIEW virtual instrument to be used to generate the Fourier series approximation of a sawtooth waveform. Note the improvement as the number of terms used in the approximation increases. Graphically compare the approximation to a noisy sawtooth waveform. Use the Fast Fourier Transform and its inverse to dissect the waveform, filter out the noise, and then reconstruct it.
Purpose
This exercise will illustrate the ability of a Fourier series to reproduce a periodic waveform and introduce you to the discrete Fast Fourier Transform (FFT) and its inverse. In addition, a simple noise filter will be implemented.

HW6: Digital Filtering
Scope
Implement a LabVIEW virtual instrument that acquires a noisy audio signal, filters the noise out, and compares the filtered and unfiltered signals.
Purpose
This exercise demonstrates how a signal can be extracted from bandlimited noise using a digital filter. Both the time domain and the frequency domain will be used to provide visualization of the signal with and without noise.

HW7: Spectrum Analysis and Aliasing
Scope
Implement a LabVIEW virtual instrument to be used to acquire a sinusoidal waveform at 1000 pps and analyze it to determine its frequency. Use a switch to toggle between a virtual signal and a laboratory signal. Illustrate aliasing by adjusting the signal frequency to purposely violate the Nyquist criteria.
Purpose
This exercise will further familiarize you with the amplitude spectrum and illustrate the aliasing phenomena. Sampling at discrete time intervals limits the maximum frequency that can be properly resolved to half the acquisition rate (*i.e.* the Nyquist frequency). Higher frequencies are not just ignored, they fold back around the Nyquist frequency and contaminate the low frequency spectrum in a process called aliasing.

PR2: Design of a DSP System
Scope
Design and implement a LabVIEW virtual instrument that acquires and decodes a seven-digit touch-tone telephone number. Decoder performance shall be verified by accepting input from one or more data files generated from the laboratory dialer.

HW8: Oversampling and Averaging
Scope
Implement a LabVIEW virtual instrument that smoothes a noisy analog signal by the technique of oversampling and averaging. Toggle between direct data and oversampled data acquired at the same effective rate. Toggle between a virtual set point with random noise added and pressure transducer acquisition.

Purpose

Unwanted noise distorts the analog signal before it is converted to a digital signal. You can minimize the effects of this noise by oversampling the signal and then averaging the oversampled points at an "effective" sampling rate. The level of noise is reduced as the square root of the number of points averaged. For example, if you average 100 points the noise is apparently reduced to 1/10 of its actual value.

HW9: Damper Motor Control

Scope

Implement a LabVIEW virtual instrument to control a Honeywell damper motor that provides a 90° shaft rotation in approximately 90 seconds for the purpose of opening and closing an HVAC damper. Toggle between the Honeywell motor and a simulated motor.

Purpose

This exercise illustrates the relationship between power supplies; digital relays, which serve as on-off switches; and actuators, which can be operated by these switches.

HW10: Dead Band Control Algorithm

Scope

Implement a LabVIEW virtual instrument to control an actuator through the use of a dead band control algorithm to minimize set point "chatter" due to a noisy signal.

Purpose

This exercise implements a strategy for controlling a noisy signal without continually "hunting" for a set point by defining a "dead band" around the set point in which no control action is taken. The controller shall seek the zero error condition when active and then remain inactive until the error exceeds the dead band tolerance.

PR3: Design of a Control System

Scope

Design and implement a LabVIEW virtual instrument that controls the air pressure in a mechanically ventilated space. Use a dialog box to choose between virtual and laboratory operation. System performance shall be verified through interaction with the Ventilation System

The Decoder Module

The decoder module provides a real-world digital signal processing example by decoding a seven-digit touch-tone telephone number. The analog data for a given phone number is acquired by the instructor through a National Instruments PCI 1200 DAQ card from a signal generated by a 33-memory tone dialer (Radio Shack PN 43-146, $24.99). Each of the 33 analog signals is written to a separate text file. The files are then provided to the students to allow them to fully test their virtual instruments without requiring a DAQ card.

Project specifications require the student to acquire 1.4 seconds of touch-tone data. The student must then decode, display, and play the telephone number through the computer’s sound card and identify the caller from a database.

The time domain data for a typical seven-digit phone number is shown in Figure 1. The automatic dialer obviously provides well-conditioned data sets that minimize the amount of code necessary to parse and decode each digit.

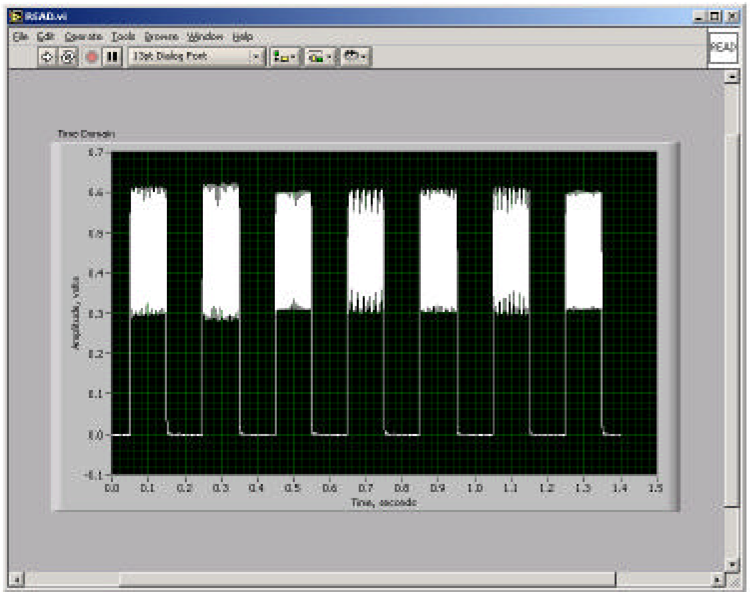

Figure 1 – Touch-Tone Data in the Time Domain

The decoder illustrates how information can be extracted from a dual-tone multiple frequency signal using bandpass digital filters. Both the time domain and the frequency domain provide visualization of a single phone digit, as shown in Figure 2.

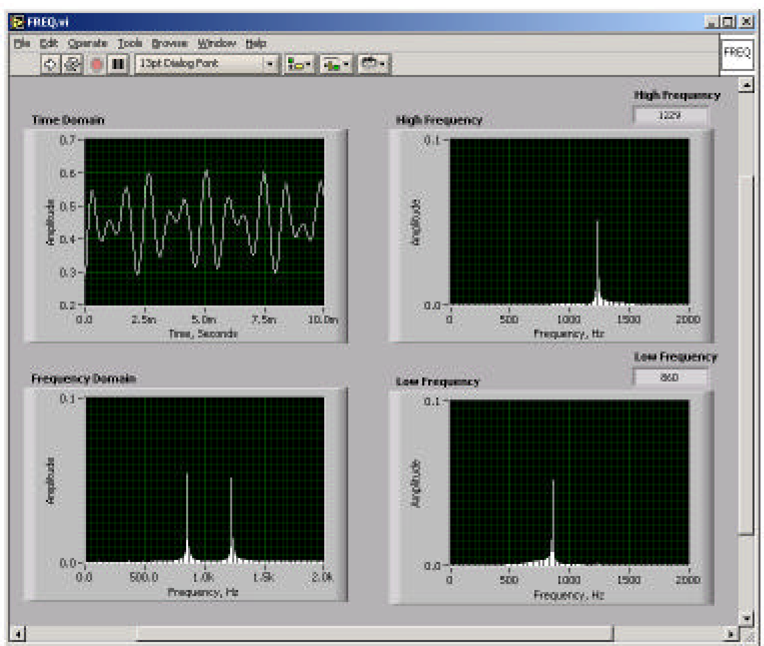

Figure 2 – Signal Visualization in the Time and Frequency Domains

The full project specifications are as follows.

(1) Create a library containing a main vi named PHONE.vi (marked top level) and all other subvi's created for this project.
(2) Create READ.vi to acquire 1.4 seconds of a composite waveform (text files provided) and display the results in the time domain.
(3) Create PARSE.vi to parse a specified individual waveform (digits 1 through 7) from the composite waveform and display it in the time domain.
(4) Create FREQ.vi to determine the high frequency (1100-1600Hz) and the low frequency (600-1100Hz) pair for a given digit from its time domain waveform.
(5) Create DECODE.vi to decode and display the name of a digit as a text character (0-9, *, #) for its specified high/low frequency pair.
(6) Create PHONE.vi to combine the above vi's to determine and display a seven-digit telephone number as a single hyphenated text string (*e.g.* 710-6830).
(7) Create ID.vi to automatically identify a caller from a telephone database (PhoneData.txt) for a given seven-digit phone number.
(8) Create BOOK.vi to display the entire telephone book (PhoneData.txt) upon demand for visual verification of a decoded phone number.
(9) Create PLAY.vi to construct a 1.4 second 8-bit mono *wav* file at 11025 pps for the seven digit phone number and play it through the sound card upon demand.
(10) Demonstrate all of the above features with a single centered and self-running Touch-Tone Decoder System managed by the main vi named PHONE.vi.

Assessment

As part of an ongoing program assessment process, students are surveyed at the end of each semester to determine their opinion of every engineering course taught at Baylor University. Their six response options are: strongly agree, agree, slightly agree, slightly disagree, disagree, and strongly disagree. Student responses for the past two semesters are shown in parentheses in the order listed above for the following three queries.

(1) Course assignments contributed to student understanding.
Spring 01 (63%, 38%, 0, 0, 0, 0)
Fall 01 (81%, 19%, 0, 0, 0, 0)

(2) Students learned a great deal from this course.
Spring 01 (63%, 38%, 0, 0, 0, 0)
Fall 01 (69%, 25%, 6%, 0, 0, 0)

(3) Professor used procedures and methods conducive to learning.
Spring 01 (50%, 38%, 13%, 0, 0, 0)
Fall 01 (75%, 25%, 0, 0, 0, 0)

The six response categories vary from left to right from most positive to most negative. As can be seen, none of the student responses were negative and most were quite positive.

As stated earlier, the goal of this course is to provide knowledge and experience in the design and implementation of computer-assisted data acquisition and computer-controlled instrumentation. Interpretation of these assessment results supports the conclusion that, at least from the student perspective, this objective has been met by the hands-on learning module approach described herein.

References

1. Bishop, Robert H., *LabVIEW Student Edition 6i*, Prentice-Hall, Inc., 2001.
2. Travis, Jeffrey, *Internet Applications in LabVIEW*, Prentice-Hall, Inc., 2000.

ROBERT T. DOTY

Dr. Doty is professor of mechanical engineering at Baylor University. He has 29 years of teaching experience and 8 years of industrial experience. He is a registered professional engineer in Louisiana and Texas. He teaches courses in design of data acquisition systems, computer-aided design, and senior design.