1996 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM
JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

INTERFACING LABVIEW WITH INSTRUMENTATION FOR ELECTRONIC FAILURE ANALYSIS AND BEYOND

Mr. Randy K. Buchanan, Lecturer
Engineering Technology Department
Pittsburg State University
Pittsburg, Kansas

KSC Colleagues - Coleman Bryan and Larry Ludwig
Material Science

Contract Number NASA-NGT10-52605

August 8, 1996

Acknowledgments

I would like to thank Coleman Bryan for providing me the opportunity to engage in the research activities involved during my summer at Kennedy Space Center. I look forward to the opportunity to work with you more in the future.

Thanks also to Larry Ludwig for sharing his laboratory and office with me, and providing direction for this research. Without Larry, the final result would have been much less prolific. I hope my anal retentiveness was not too much of a nuisance.

Several people were very helpful throughout the summer in various aspect of providing support. Thanks goes to Kurt Leucht for all his computer and network support, and to Jeff Rauwerdink for helping to establish a "place" for me, as well as daily diversion.

Andy Finchum and Jon Bayless were instrumental in acquainting me with various areas of the laboratories and providing me with my first few days of entertainment. Thanks to Andy for your many personal tours of places where we probably should not have been -- they were the best. Thanks to Jon for supporting my efforts and being my first guinea pig (that's a compliment).

Steven Huff was also a help in day-to-day tasks, but was especially helpful in aiding me with visuals for my mid-term presentation. I had no idea graphics files could be so large.

Thanks to Bill Dearing for computer support and for servicing my E-mail account, sometimes on a daily basis. He was gracious in accepting that I wouldn't be working in his lab after my arrival.

Thanks to Farshid Sepehri, graduate student, for assistance during the first half of the summer.

Thanks to Gregg Buckingham for all your special effort in making this summer such a dynamic experience for all involved. Your work and efforts are noticed, and appreciated. Also thanks for taking the time to meet with me and my student.

Thanks to Kari Stiles and Richard Johnson for their summer-long efforts and behind the scenes coordination. The resultant program you helped produce was both informative and enjoyable.

Author Notes

Preparation of this manuscript was supported in part by a NASA Faculty Summer Fellowship sponsored by the University of Central Florida and the American Society for Engineering Education. The tutorial guide referenced throughout this paper is available directly from the author or from Larry Ludwig, NASA/LO-MSD-2E, Electronic Failure Analysis Laboratory, NASA Kennedy Space Center, FL 32899. Correspondence should be addressed to Randy Buchanan, Department of Engineering Technology, Pittsburg State University, Pittsburg, KS 66762, or call (316) 235-4370.

Abstract

The software, **Lab**oratory **V**irtual **I**nstrumentation **E**ngineering **W**orkstation (LabVIEW), is designed such that equipment and processes related to control systems can be operationally linked and controlled by the use of a computer. Various processes within the Failure Analysis Laboratories of NASA Kennedy Space Center demonstrate the need for modernization, and in some cases automation, using LabVIEW. An examination of procedures and practices within the Failure Analysis Lab resulted in the conclusion that some device was necessary to elevate the potential users of LabVIEW to an operational level in minimal time. This paper outlines the process involved in creating a tutorial application to enable personnel to apply LabVIEW to their specific projects. Suggestions for furthering the extent to which LabVIEW is used are provided in the areas of data acquisition and process control.

INTERFACING LABVIEW WITH INSTRUMENTATION FOR ELECTRONIC FAILURE ANALYSIS AND BEYOND

Randy K. Buchanan

1. INTRODUCTION

Various processes within the Failure Analysis Laboratories of NASA Kennedy Space Center demonstrate the need for modernization, and in some cases automation. The commonalty which exists with all processes is the need for computer control, either in the form of improving existing hardware and software, or the creation of computer control where none has previously been used. Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) software has been indicated by the Failure Analysis Laboratory to be a desirable solution to accomplish most of these tasks. In addition, LabVIEW is widely accepted in industry as a premier solution to facilitate or enhance computer control processes. Standardization utilizing this software would place the lab in a state-of-the-art continuum with leading technology in the field, allowing for quick/flexible control with the ability to easily adapt to changes and advances.

Processes targeted for modernization included the Slack Vacuum Chamber, Electrostatic Discharge Chamber, High-Amperage DC Test Fixture, High-Voltage AC Test Fixture, Electronic Failure Analysis Test Fixture, and any custom configurations for units under test. During initial investigations into these processes, it became apparent that the gap between LabVIEW programming and implementation must be bridged before modernization initiatives could be seriously undertaken.

2. BACKGROUND AND SIGNIFICANCE

LabVIEW is a powerful and flexible instrumentation and analysis software based on the programming language "G". The programming language "G" is in some ways similar to common languages, like C or BASIC, in that it is general purpose with extensive libraries of functions for any programming task including conventional program development tools. It is different however, in that it is graphical in nature. In addition, it includes extensive libraries of special functions, data acquisition, General Purpose Interface Bus (GPIB) and serial instrument control, data analysis, data presentation, and data storage. (ref. 1) It is one of the few languages that is platform independent, meaning that the same program can function in Windows on a PC, and the operating systems of UNIX, Macintosh, and Sun. Modular programming is also a feature which makes LabVIEW so attractive. Programs may be created with segments, much like sub-routines in a traditional language, in any order, and compiled into a singular high level program which functions as one. A unique characteristic of LabVIEW is its data-flow principle of processing inputs and outputs. Whereas typical languages use control-flow commands and a program counter to sequence through programming steps one at a time, LabVIEW processes input data as soon as all data is received at the proper input node. More than one data line may be processed at any given instant in time, as shown in Figure 1.

One should note that LabVIEW is not limited to General Purpose Interface Bus (GPIB) type communication. It is designed to operate over VME Extensions for Instrumentation (VXI) and serial mediums, as well as digital and analog data acquisition and process control systems.

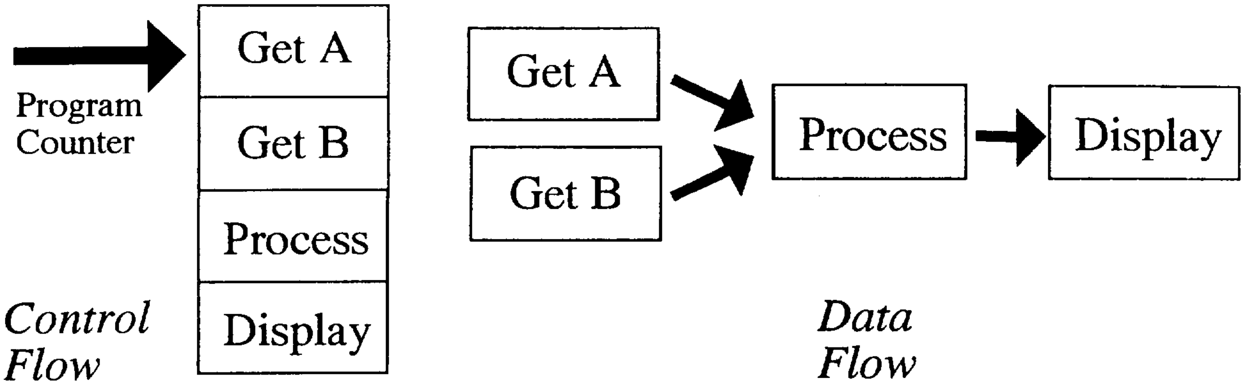


Figure 1. LabVIEW Data-Flow Principle (ref. 2, p. 78)

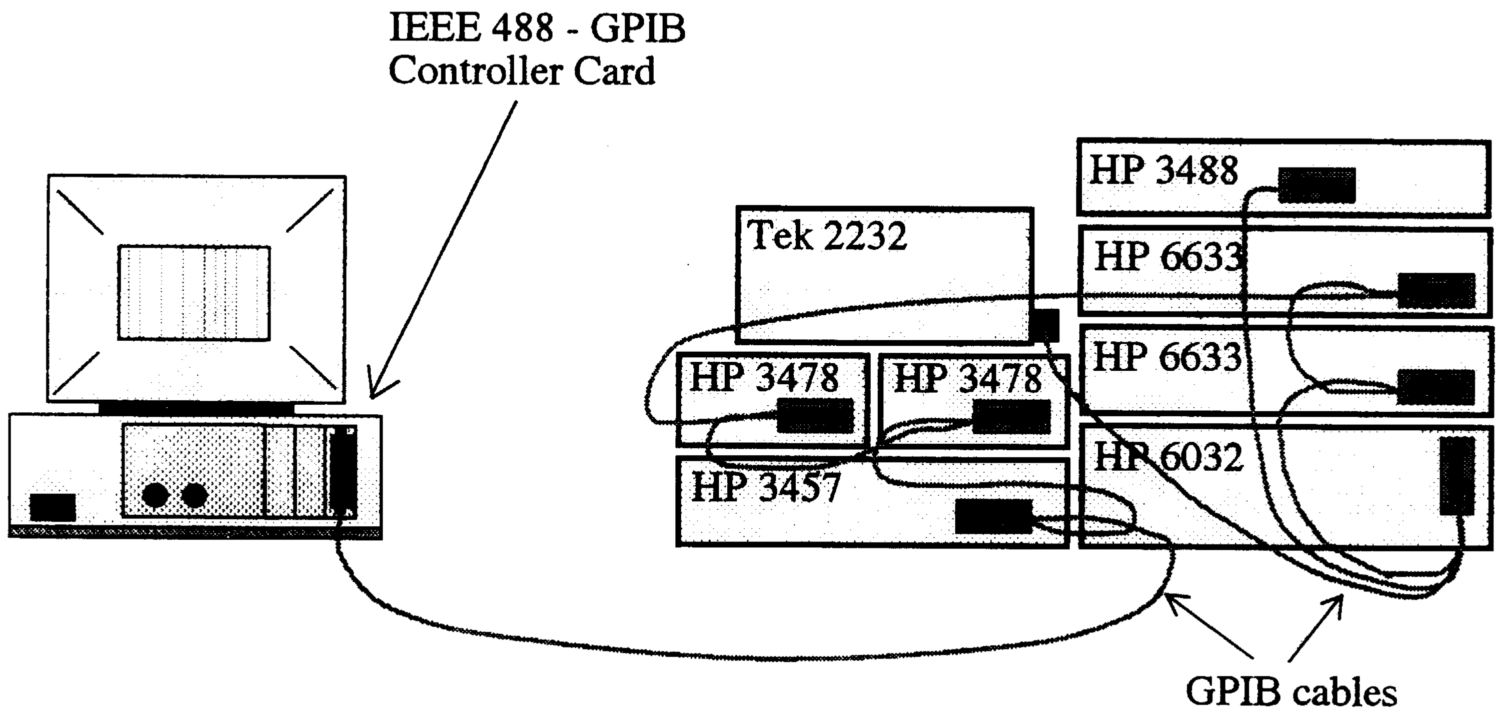
Many problems surround the implementation of a software such as LabVIEW. Learning curves for new software can be fairly flat, especially in the beginning. The time required to learn new software removes the engineer from the daily tasks at hand. Often, it is the case that the engineer will delay committing the time to learn the new software in lieu of work responsibilities. An examination of procedures and practices within the Failure Analysis Lab resulted in the conclusion that some device was necessary to elevate the potential users of LabVIEW to an operational level in minimal time. Thus became the scope of my fellowship.

Search of available literature and documentation revealed that there exists a lack of available training in the areas of LabVIEW implementation. In only one instance was it found that an attempt had been made to initiate software manipulation for simplification of LabVIEW software/hardware utilization. However the scope of this project, which was performed by a private contractor, is limited as its results have not been made available for public consumption. (ref. 3) Opportunities exist for training in terms of LabVIEW programming directly from National Instruments, the maker of LabVIEW. Training for implementation of the software with physical instruments, however appears to be non-existent. Therefore, it was deemed necessary to devise a method by which all lab personnel would be able to acquire the tools necessary to create applications using LabVIEW in each of their own areas. Ultimately, a step-by-step instruction reference procedure, designated "Tutorial Guide for Interfacing LabVIEW with GPIB Addressable Instrumentation" (unpublished guide, see Author Notes) was created to achieve this result. In addition, a computer with GPIB and LabVIEW capability was established with 8 GPIB addressable instruments, in an arrangement conducive to training and instruction.

3. APPARATUS

For the training apparatus, instrumentation operation and control was facilitated via the GPIB standard, while coalesced with a computer station possessing an IEEE 488.2 interface board, and operated under a LabVIEW environment. The Electronic Failure Analysis (EFA) Test Fixture (see Figure 2) was chosen as the system for the training apparatus. This test fixture was deemed to possess the most instrumentation which was common to all areas of the lab, especially in the Electrical/Electronic Failure Analysis area. More importantly, however, it was deemed that the EFA would be the most user-friendly system from which to learn.

The EFA Test Fixture was altered to support LabVIEW and GPIB communication. Equipment was coupled via GPIB parallel cables, from the IEEE-488 card in the computer, and sequentially to each piece of equipment in the fixture, as indicated in Figure 2.



Computer with LabVIEW

Gateway 2000 P5-60 Computer

National Instruments AT-GPIB/TNT
IEEE-488.2 Controller Card

LabVIEW for Windows Version 3.1

GPIB Instruments

Tektronix 2232 100 MHz Digital Storage Oscilloscope
Hewlett Packard 3478A Digital Multimeter (2)
Hewlett Packard 3457A Digital Multimeter
Hewlett Packard 3488 Switch/Control Unit
Hewlett Packard 6633A System DC Power Supply (2)
Hewlett Packard 6032A System Power Supply

Figure 2. LabVIEW and GPIB Instrumentation Interfacing Utilizing the Electronic Failure Analysis Test Fixture

Specific ordering of connections to the GPIB cables is not important, as long as all equipment is connected at some point to the GPIB "bus".

LabVIEW Virtual Instrument (VI) programs which are designed to operate specific equipment were obtained from National Instruments and modified or re-configured for the test fixture instruments. (for instructions refer to unpublished guide, see Author Notes) Most VIs were to designed to consume most, if not all, of the visible area of the computer monitor display screen. Since the training apparatus included 8 different instruments, a method of incorporating all instrument VIs into a single screen VI was necessary. Therefore, several VIs incorporating all 8 instruments were developed for the EFA fixture. Multiple VIs were made to suit users' various preferences.

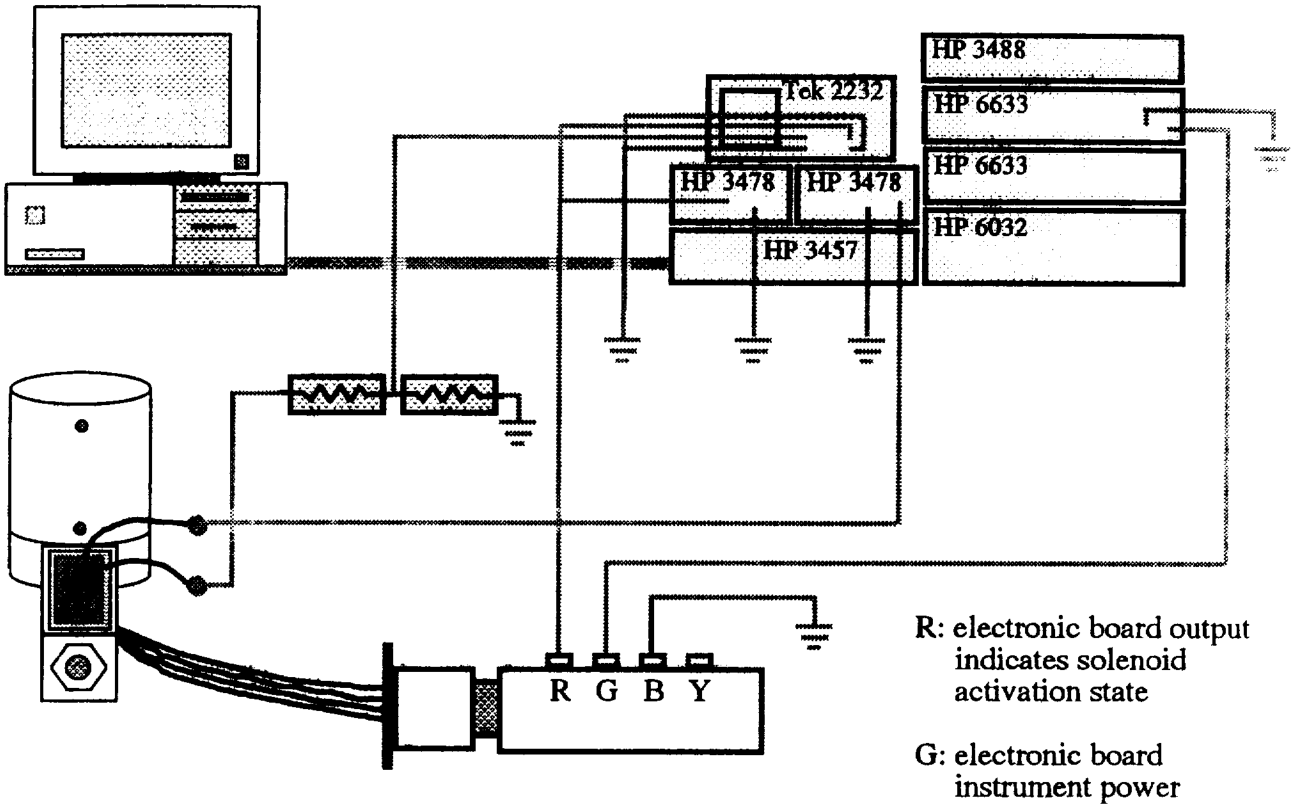
Finally, GPIB communication was established, and all instruments verified for GPIB operation via LabVIEW. Hardware and software issues were resolved as necessary, and procedures for solving issues not specific to this particular fixture, were included in the tutorial guide (unpublished guide, see Author Notes).

4. TESTING

Once each instrument VI was configured to operate within the existing system, they were individually executed to assure proper operation. Errors which occurred, if any, were responded to and appropriate actions taken to resolve the problem. All individual VIs within the EFA user VI were initially tested for proper operation by first physically connecting the output of each of the three power supplies to the input of one of the three multimeters. The oscilloscope was also connected to one of the power supplies to provide an input for measurement. All VIs were concurrently ran after adjusting the power supplies for the desired voltage output. Upon program completion, the EFA user panel multimeter outputs were read to verify that their voltage measurements concurred with original power supply settings.

It was then important to provide a set-up which lab personnel could relate to, as well as train with. To provide for training on the test fixture, EFA employees were consulted for suggested training environments. Per recommendations, a unit-under-test arrangement was constructed involving a Marotta Solenoid Valve. The test Fixture was designed to provide a means to analyze the functioning of the valve, as well as to study GPIB control. The solenoid was powered from a 0 V resting position to a 28V dc activated state. The response voltage waveform was observed by connecting the oscilloscope to a resistor network as shown in Figure 3.

Specifications for the unit-under-test Marotta solenoid valve with a magnetic armature reference are as follows. The valve position is read by an onboard electronics module which provides a 28 volt output for an activated valve. The electronics module uses a Hall Effects Device (HED) sensor to activate a transistor driven output stage, providing either a 0.6 volt (or less) output or a 28 volt dc output according to the valve position. The two most important electronic valve parameters are the current waveform through the solenoid valve coil and the electronics module output waveform. From the current waveform the actual activation time of the valve can be determined, including the duration of armature movement. The output waveform from the electronics module, when captured at the same time as the current waveform, shows the delay time between armature movement and the electronics package response. It also shows whether the electronics package is providing a correct output per the valve armature's actual position. Operations and measurements were verified to be correct between LabVIEW and the actual instrument readings.



R: electronic board output
indicates solenoid
activation state

G: electronic board
instrument power

Figure 3. EFA Marotto Solenoid Valve Test Fixture

5. TRAINING

With the test configuration in place, real training opportunities for EFA employees to utilize LabVIEW in needed control systems is available. To this end, the EFA LabVIEW and GPIB Instrumentation Training Configuration has been placed into operation, and the tutorial guide (unpublished guide, see Author Notes) has been submitted for use by the Failure Analysis and Physical Testing Branch.

The tutorial guide provides background information into GPIB, including controller card requirements and system cable connection. Sources for VI acquisition are included, and most importantly, the LabVIEW File Transfer Protocol (FTP) Internet address for VI downloading is provided. Most of the step-by-step procedures outlined in the guide relate to creating customized application-oriented VIs. These procedures include instructions for creating both the user panel, which is visible to the operator via the computer screen, and the user diagram, or program, which actually enables the panel and the VI as a whole to function as desired. Illustrations of the relevant panels and diagrams are included to increase understanding and to provide a visual guide through the process.

Many opportunities exist for personnel to use the tutorial guide in their specific areas, whether or not related to failure analysis. The guide may be copied and distributed to the appropriate parties outside the EFA. The steps outlined in the tutorial deal with creating a method by which to facilitate the interfacing of LabVIEW with GPIB instruments, not with failure analysis itself. Essentially, the programming and communication methods are the same, regardless of the type of instrument which is being implemented.

The tutorial guide has already been utilized by laboratory personnel, with positive results. A member of the Electronic Failure Analysis Lab was successful in following the tutorial instructions and creating an application in a matter of hours. The process included downloading VIs from the Internet, re-configuring to fit his application, and programming a user interface panel which incorporated all the desired instruments into one panel. This particular engineer had no formal training in the use of LabVIEW previous to using the tutorial.

Much can be said as to the advantages of having hardware to "see and touch", when attempting to enlighten oneself on methods of programming and implementation. A system such as the Electronic Failure Analysis LabVIEW and GPIB Instrumentation Training Configuration provides personnel with an opportunity to train with actual instrumentation before them. The Marotto Solenoid Valve Test Apparatus was devised to provide a set-up which Electronic Failure Analysis Laboratory personnel could relate to, as well as train with. Similar relative apparatus could be established in other areas to provide motivation and a method to conduct training specific to an area. This not only holds greater interest, but also provides instant visual feedback as a method of indicating whether certain operations would or would not be successful.

6. DISCUSSION

Establishing a dependable reference test station requires alleviation of all real and potential problems. Implementation of the Electronic Failure Analysis LabVIEW and GPIB Instrumentation Training Configuration required a test of all associated wiring. Power supply panels, which provide voltage to the test fixture via voltage buses, required re-wiring to attain a standardized polarity and ordering in the form of a predictable formation. All GPIB cables were installed or re-routed to achieve reasonable use from the various cable lengths involved. Eventually one GPIB cable was found to be defective and was consequentially replaced. As such, physical communication interfaces should be established before investing time in the programming side of implementation.

Like any other operation which involves a computer programming, if all parameters are not met or are declared incorrectly, the system will not operate properly. Therefore, it is important to take time to learn of the details necessary to make the system work right the first time. It is fortunate, however, that LabVIEW tends to be much more forgiving than traditional programming languages, in the sense that many times it will help you locate the problem. Often it appears that it has foreseen the mistakes you may make and ultimately contains a built-in remedy. LabVIEW has a unique method of troubleshooting by which it steps though your program (even if it is not written sequentially) and visually indicates the numeric or string values of each node. This, as well as other features makes programming in LabVIEW enjoyable in comparison.

Although GPIB is the target medium of communication for implementing LabVIEW in this case, it is not the only medium available. Specific procedures outlined in the tutorial guide correlate with GPIB instrumentation. Similar methods of implementation could, however, be utilized for other environments. In fact, this GPIB configuration could be used as a model for development of other systems with minimal difficulty.

Limitations to the use of LabVIEW within a control system, such as for process control, would include the application to high-speed operations. Typically, internal timing functions should be limited to 0.1 second. (ref. 2, p. 31) This timing constraint varies between applications and depends upon the characteristics and type of platform (PC, Macintosh, etc.) which is being used. External "smart" controllers may be used to all but alleviate the timing issue. In such a system the controller assumes control responsibilities and LabVIEW provides the Man-Machine Interface (MMI).

6. CONCLUSIONS AND RECOMMENDATIONS

LabVIEW provides a solid yet flexible foundation for configuring computer controlled instrumentation. Instrument control using LabVIEW as covered is the fundamental tool necessary for various levels of testing and analysis. The tutorial guide developed for this purpose will enable laboratory personnel to create LabVIEW applications in less time than previously possible, by providing step-by-step instructions for acquisition, augmentation, and implementation of virtual instrument programs. The EFA fixture will provide the hands-on experience necessary for staff training and development, by serving as a model for others to emulate. Although the tutorial and EFA test fixture were designed for the Electronic Failure Analysis area, their application and use should not be limited to such. Procedures dealt with relate to the implementation of LabVIEW with instrumentation, not with failure analysis itself.

Beyond the realm of instrument control, many situations involve controlled and manipulated variables. LabVIEW-linked instrumentation, to the extent explored in this investigation, may be used to aid in the measurement and manipulation of some of these variables, but would fall short of providing full control. The inclusion of additional plug-in boards and/or the cooperative use of external controllers, would provide the ability to use LabVIEW to effect control over a particular process. Therefore, it is recommended that similar training in the areas of data acquisition and process control be developed.

REFERENCES

[1] National Instruments. LabVIEW Graphical Programming for Instrumentation: User Manual for Windows. National Instruments Corporation; Austin, TX; 1992.

[2] Johnson, Gary W. LabVIEW Graphical Programming: Practical Applications in Instrumentation and Control. McGraw-Hill, Inc.; New York; 1994.

[3] I-NET. Generic LabVIEW. KSC-11703 Kennedy Space Center Report; year unknown.