Graphical Programming Tools for Electrical Engineering Higher Education

doi:10.3991/ijoe.v7i1.1403

E. Lunca[1], S. Ursache[1] and O. Neacsu[1]
[1]Technical University of Iasi, Romania

***Abstract*—As the engineering education and computer technologies for industry merge, the opportunities for teaching and learning will also expand. In such context, a large number of academic courses that teach electrical engineering concepts have incorporated the graphical programming as educational tool for student use. The current paper discuses some of the most popular graphical programming packages, identifies two alternative solutions and shows how the virtual instrumentation approach pioneered by LabVIEW may enhance the electrical engineering curricula.**

***Index Terms*—graphical programming, virtual instrumentation, electrical engineering (EE), higher education.**

I. INTRODUCTION

As the graphical programming has grown in popularity, its usage in the educational field has become an important preoccupation for more and more universities all over the world. In electrical engineering, graphical programming environments like NI LabVIEW and Agilent VEE have now the potential to enhance the content of a large variety of courses [1-14], offering teachers the opportunity to develop new educational tools in the form of "virtual instruments".

By integrating PCs and commercial technologies, the virtual instrumentation approach is ideal to implement software-based versions of the real-world instruments, increasing the versatility of the existing hardware with minimal cost. This, in combination with the intuitive nature of the dataflow programming, allows educators to create a richer learning environment that greatly facilitates the hands-on experimentation. Likewise, the students may develop their own applications without needing extensive knowledge of programming techniques, often associated with text-based engineering software. Thus, they will be able to focus on exploring concepts and solving engineering problems, rather than learning programming nuances.

II. POPULAR GRAPHICAL PROGRAMMING TOOLS FOR ELECTRICAL ENGINEERING

Because of the advantages of graphical programming, a range of software tools for different areas of electrical engineering have been developed by industry. As a natural result, some of these products started to play an important role in the EE higher education, as well.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is an industry-standard, fully-featured graphical programming language, offered by National Instruments since 1986 and extensively used for developing test, control and measurement applications. Because LabVIEW dramatically simplifies and accelerates the design projects, the educators have incorporated LabVIEW in a large variety of courses, from introductory classes to graduate-level studies. It can be used both in classrooms (or laboratories), for in-class concept demonstrations and prototype implementation, and in homework, to compute, simulate and identify solutions to engineering problems. Note that, because the appearance and operation of the LabVIEW programs imitate physical instruments, such as oscilloscopes or multimeters, they are called virtual instruments or VIs.

The LabVIEW programming is extremely intuitive. Disposing of various objects, such as terminals, constants, structures, functions and subVIs (sub-virtual instruments), LabVIEW allows users to define the behavior of their applications by dragging and dropping these objects onto a diagram, and connecting them through wires, as necessary. Each terminal in the block diagram has a correspondent on the front panel window (the graphical user interface of the virtual instrument), which acts either as a data input or a data output. Generally, this programming approach is very well received by students, many of them stating that the LabVIEW platform is highly accessible and efficient for learning.

LabVIEW comes with hundreds of ready-to-use example VIs. Moreover, it integrates configurable Express VIs that significantly reduce the development time and complexity associated with adding analysis and signal processing algorithms into user applications. Finally, instructors and students may save time and acquire new ideas by visiting the National Instruments' academic web site, where they can access labs, exercises and tutorials covering a large number of electrical engineering disciplines.

Agilent VEE (originally known as HP VEE) is a "visual engineering environment" designed to quickly develop and automate measurements and tests. It is characterized by its ability to control a large variety of instruments, as well as its focus on high-level task-oriented blocks.

Programming with VEE also requires creating block diagrams, but the VEE programs do not necessitate implementing front panels. However, if a user interface has to be developed, this can be done by simply adding objects and "UserFunctions" to a front panel.

As in the case of LabVIEW, Agilent VEE comes with built-in tutorials and sample programs. Both environments have student editions at affordable prices, intended to help students in designing graphical programming solutions to their classroom problems and laboratory experiments. Academic programs for universities aimed to help educators to teach these two software platforms and to include them in course curriculum are also available.

DASYlab is an "icon-based data acquisition, graphics, control and analysis software" developed by DasyTec. In DASYlab, module icons are placed on a worksheet window and connected with wires in a schematic diagram, which defines the flow of data through the system. Each icon in the diagram represents an input, operation or output function.

The DASYlab flowcharts have an appearance similar to the LabVIEW block diagrams, but there are some underlying differences between the graphical methodologies used by these two environments. National Instruments, which now owns both products, states that "although LabVIEW does not provide as much automatic data handling of measurement information as DASYLab, it contains a more flexible and powerful programming interface that allows you to define advanced and complex programs quickly" [15]. Anyway, since DASYLab has a very short user-learning curve, it can be easily integrated in any measurement laboratory.

Snap-Master (from HEM Data Corporation) is a complete software package designed to quickly acquire test data. It uses an intuitive flowchart approach based on icons (specific functions) connected through "data pipes", which defines the data flow between these functions. Simply double clicking on an icon brings up a dialog box or table containing all the settings for that function.

The product philosophy of Snap-Master lies in three "no programming" modules (Data Acquisition, Waveform Analyzer and Frequency Analyzer), which can operate as stand-alone packages, or integrated into a complete data acquisition and analysis solution. Additionally, add-on programming modules are available for extending its capabilities beyond the data acquisition solutions whenever necessary.

Because Snap-Master is very simple to use, it can be integrated in the educational laboratories carrying out measurements, as well. An attractive feature of Snap-Master consists in its "Sensor Database", which automatically converts acquired data values to engineering units, such as rpm, psi, degrees, etc.

Simulink, developed by MathWorks, is a graphical programming environment for multidomain simulation and model-based design of dynamic and embedded systems. By offering a "block diagramming" approach and a customizable set of block libraries, it allows users to design, simulate, implement and test a diversity of time-varying systems, including communications, control, signal processing, video processing and image processing. Also, it is worthwhile to remark that Simulink is fully integrated with MATLAB, providing instant access to an extensive range of tools for developing algorithms, analyzing and visualizing simulations, creating batch processing scripts, customizing the modeling environment, and defining signal, parameter and test data.

In education, the MATLAB / Simulink software and add-on products are used now for enhancing the teaching activities in many fields, from engineering to science and finance. Except a few small differences, the student versions of these two products provide all the features and capabilities of the professional versions at very competitive prices. Free online curriculum kits including technical literature and demonstrations, tutorials, webinars, suggested textbooks and homework problems for students are provided by MathWorks to help educators in planning and teaching MATLAB / Simulink related courses. Furthermore, a large volume of other educational resources, such as downloadable code and models, is also available for classroom instruction or individual learning.

At the moment, the MATLAB / Simulink software can be linked to many other commercial products, including some of those presented in this paper. For instance, through the LabVIEW Simulation Interface Toolkit, it is possible to build custom user interfaces for models created in the Simulink environment, while using the LabVIEW Math Interface Toolkit allows calling LabVIEW VIs directly from the MATLAB environment. Similarly, built-in MATLAB Script objects, which provide access to the functionality of MATLAB, are available for Agilent VEE users (the VEE Student Edition requires MATLAB to be installed on the user machine).

III. ALTERNATIVE GRAPHICAL SOLUTIONS

Until now, a number of software products that use a graphical programming approach were presented. However, there are graphical software packages based on different approaches, intended to eliminate or make transparent to users the programming aspects. Two such products are briefly discussed below.

DADiSP (pronounced *day-disp*) is a visually oriented software package for displaying, managing, analyzing and presenting scientific and technical data, acting as an "interactive graphics worksheet". A DADiSP worksheet is comprised of analysis windows that contain either raw data or data transformed by one of the analysis functions of DADiSP, which are shown as graphs or tables. Through formulas, the data or graph in a window can be related to those in other windows, permitting users to define their own analysis chain without programming. When new data is loaded into a raw data window, all the dependent windows will automatically recalculate and update.

For education, a great advantage of DADiSP consists in its free student edition – a 9-windowed, menu-driven spreadsheet that accepts data series of up to 32768 points. By using pop-up menus and icon buttons, the students can quickly build sophisticated analysis chains, incorporating routines for signal arithmetic and calculus, signal editing, waveform generation, peak finding, Fourier analysis, matrix manipulation, graphing and plotting, custom menu creation, text and annotation.

TestPoint, a trademark of Capital Equipment Corporation, is a software tool that uses object-oriented, drag-and-drop technology for designing test and measurement applications. Working with TestPoint requires starting with simply dragging custom and standard objects (buttons, graphs, switches, interface hardware, math formulas, etc.) into application, followed by dragging the objects to an action list and choosing actions from a menu, to tell TestPoint the steps involved in carrying out the desired task. In response, the software types action lines that describe the test in a clear, readable format.

While dragging and dropping objects to the action list, TestPoint creates a compiled program. The action lists support sequential execution, loops and conditionals, as well as the capability to respond to events and interrupts and to support multitasking. In addition, the TestPoint action lists can activate other action lists, to provide fully modular programming [16].

TestPoint has an academic version, too. A useful feature for quickly implementing lab applications is the "Test Wizard", which allows students to choose among pre-built application templates in a palette of graphical styles. With just a few clicks of mouse, they can create a complete running data acquisition program.

IV. SOME CONSIDERATIONS FOR USING LABVIEW IN TEACHING ELECTRICAL ENGINEERING CONCEPTS

Working with both hardware and software for research and teaching, especially in the instrumentation area, we found LabVIEW to be a powerful tool for developing new educational resources. Therefore, many of our laboratory applications were designed as PC-based experiments, controlled by LabVIEW programs. As we'll discuss further below, this environment provides all the means for solving the most common problems encountered in a typical EE laboratory: cost, development time, hardware compatibility, student interaction, etc.

A. *Cost*

As mentioned earlier in this paper, a LabVIEW student version is offered by National Instruments (the current version is 2009). Intended for educational purposes only, it has all the features of the LabVIEW Full Development System, with one difference – the addition of a watermark that appears on any printout. Together with a number of specific modules and toolkits, the LabVIEW Student Edition can be acquired for less than $100. If integrating it with low-cost data acquisition hardware, such as NI-USB 6008/09 multifunction DAQ products (Figure 1), a broad range of laboratory experiments may be implemented at an attractive overall price. From simple applications like monitoring the output of a sensor or displaying the waveform of a function generator to more complex applications requiring additional components and circuitry like controlling a small DC motor, all are possible with such a combination.

B. *Development time*

There is no doubt that the graphical programming approach of LabVIEW greatly reduces the development time. However, configuration-based development and code-generation tools, such as Express VIs and measurement assistants, reduce even more the time spent to create user applications. For instance, LabVIEW 7 Express that we currently use in the teaching activity integrates more than 40 Express VIs, which encapsulate functionality for the most common measurement and automation tasks.

Figure 2 shows the front panel and block diagram of a virtual instrument that measures the frequency spectrum and total harmonic distortion (THD) of the low-frequency signals connected to a NI-USB 6009 data acquisition board. As it can be easily seen, this program requires very little programming effort thanks to the use of three Express VIs: one for signal acquisition, one for computing THD and one for computing the frequency spectrum. Simply placing any of these functions on the block diagram will open a configuration window allowing users to choose between different options. For example, the *Spectral Measurements* VI provides the possibility to select the measurement type (magnitude – RMS, magnitude – peak, etc.), result (linear or dB), FFT window function (None, Hanning, Hamming, Blackman-Harris, etc.), averaging (together with its properties) and some phase options.

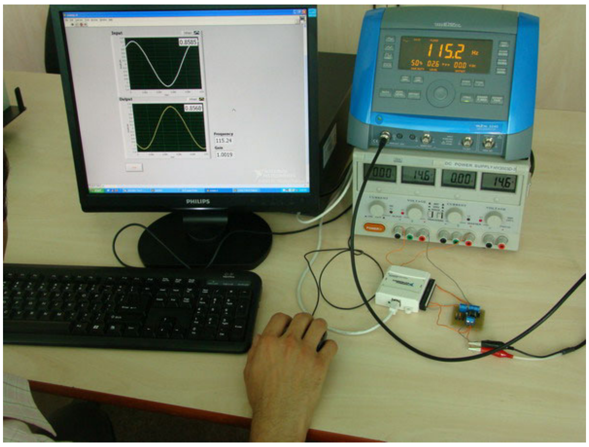


Figure 1. In this example, LabVIEW Student Edition and NI-USB 6009 are used by students to experiment with amplifying circuits

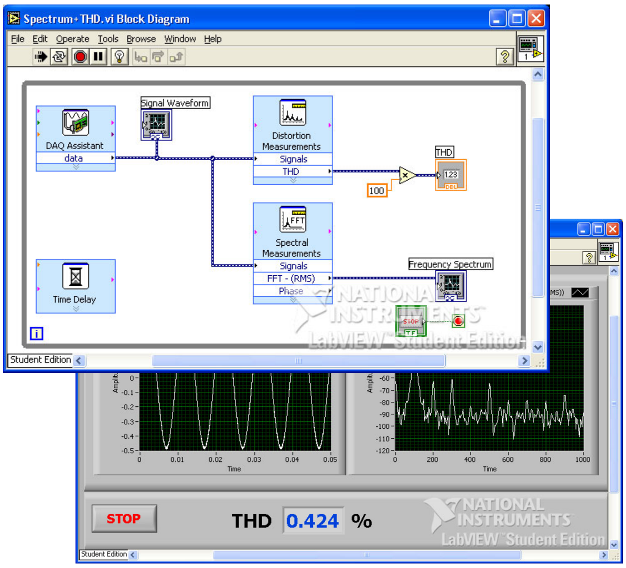


Figure 2. LabVIEW program based on Express VIs, used by students to measure the frequency spectrum and THD

Besides Express VIs, LabVIEW also integrates a large number of ready-to-use example VIs. Signal analysis and processing (distortion measurements, FFT and frequency analysis, time-domain analysis, level measurements, conditioning and filtering, etc.), hardware input and output (DAQmx, GPIB, serial, VISA, etc.), networking (DataSocket, TCP-IP and UDP, Internet and Web, etc.) are just a few categories of built-in example VIs that may serve as pedagogical tools for electrical engineering courses. Even when there is no access to real instrumentation, students can perform realistic experiments thanks to the extensive collection of simulation VIs, designed to become familiar with various LabVIEW features or to demonstrate how to solve specific engineering problems. Moreover, since the code of any example VI is completely accessible to users, it may be easily incorporated in new applications, and, therefore, the development time will decrease significantly.

C. *Hardware compatibility*

A key benefit of LabVIEW over other development environments is the extensive support for accessing measurement and test hardware. In addition to NI drivers like

NI-DAQmx, NI-488.2 or NI-IMAQ, more than 8000 drivers for instruments coming from over 275 third-party vendors are available online through the Instrument Driver Network. Since downloading these drivers is free of charge, they greatly enhance the instrument integration and measurement automation process within the university laboratories.

If necessary, the users can develop their own drivers by utilizing VISA [17], which is the standard API (Application Programming Interface) throughout the instrumentation industry. It provides interface independence, so that GPIB, serial, Ethernet, USB or VXI equipments can be programmed by learning a single set of functions that work for any type of interface. However, rather than creating instrument drivers, which is a time-consuming activity, the educators and students may use the VISA functions in LabVIEW to write simple programs that accomplish only the required functionality. Such an example is shown in Figure 3, where a magnetic field meter, manufactured in our laboratory, is remotely controlled via RS-232.

D. *Software add-ons*

From the discussion so far, we have observed that LabVIEW supports modules and toolkits that extend its graphical programming for specific applications. As presented by National Instruments on its website, there are more than 25 add-on products classified into six broad categories [18]:

- *Embedded Design*: design, prototype and deploy embedded applications to a variety of processing targets, including real-time and FPGA-based systems as well as custom microprocessor and microcontroller devices;
- *Control Design and Simulation*: combine algorithm development, analysis and visualization in LabVIEW with tools for system identification, control design, simulation and implementation;
- *Image and Signal Processing*: incorporate hundreds of image and signal processing functions into LabVIEW applications;
- *Industrial Monitoring and Control*: deploy LabVIEW to networked programmable automation controllers (PACs) for creating reliable distributed monitoring and control systems, and connect to existing programmable logic controllers (PLCs) and enterprise systems;
- *Software Development and Deployment*: develop and deploy professional, higher-quality LabVIEW applications using a variety of software engineering tools;
- *Report Generation and Data Storage*: export acquired data to third-party software tools, such as Microsoft Office and industry-standard databases, and publish information to the Web.

Because the LabVIEW Student Edition Software Suite contains the most popular LabVIEW add-ons, it may serve as a low-cost solution for teaching concepts within multiple courses, including Sensors and Instrumentation, Digital Signal Processing, Digital Communications, System Identification, Intelligent Control Systems, Machine Vision, etc. The content of this package and its applicability (according to the classification above) are shown in Figure 4.

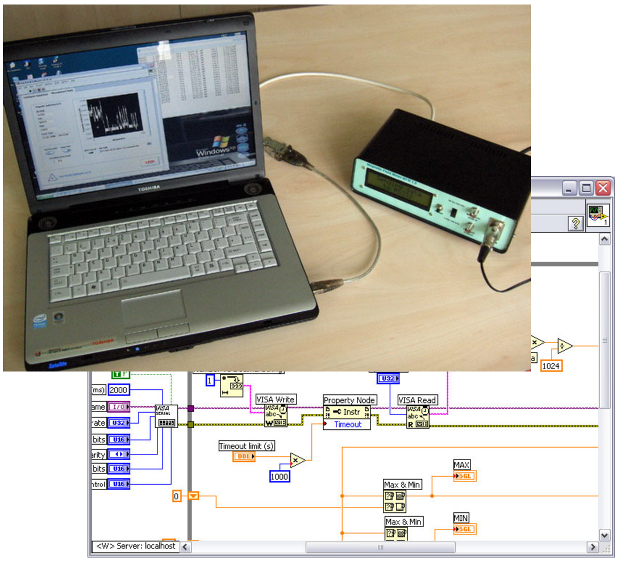


Figure 3. Using VISA to easily communicate with a magnetic field meter via the RS-232 interface

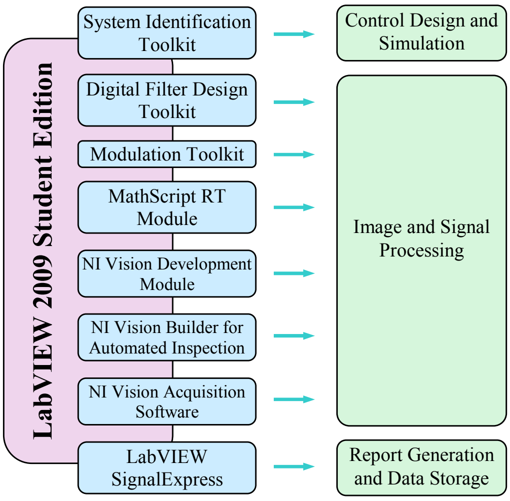

LabVIEW 2009 Student Edition
System Identification Toolkit
Digital Filter Design Toolkit
Modulation Toolkit
MathScript RT Module
NI Vision Development Module
NI Vision Builder for Automated Inspection
NI Vision Acquisition Software
LabVIEW SignalExpress
Control Design and Simulation
Image and Signal Processing
Report Generation and Data Storage

Figure 4. Modules and toolkits contained by the LabVIEW Student Edition Software Suite and their applicability

E. *Experiment reconfiguration*

Very often, reconfiguring or modifying LabVIEW-based experiments requires only acting on pieces of software code, without needing to use new circuits and hardware. As consequence, instrument clusters for teaching concepts within different courses can be configured on the same system. Lowering the costs and reducing the space requirements associated with traditional instruments will be direct results.

A flexible platform for instrumentation, design and prototyping is NI ELVIS (NI Educational Laboratory Virtual Instrumentation Suite) – our choice for many educational projects. Based on LabVIEW, it uses a multifunction DAQ device and a benchtop workstation to provide the functionality of a complete suite of laboratory instruments: oscilloscope, digital multimeter, function generator, variable power supply, Bode analyzer, dynamic signal analyzer, etc. In addition, a prototyping board that plugs into the benchtop workstation provides an area for building electronic circuitry and allows the connections necessary to access signals for common applications [19]. Note that NI ELVIS II/II+, the latest products in the NI ELVIS family, support auxiliary boards (Figure 5), which allow teaching multiple disciplines:

- the Emona DATEx Telecommunications board provides students with more than 30 modern digital and analog communications experiments in a single-board trainer;
- with the Freescale HCS12 Microcontroller Student Learning Kit and NI ELVIS, it is possible to access a complete toolchain for designing, prototyping and deploying microcontroller experiments;
- the Quanser Engineering Trainers line of boards (QNET) is designed to teach control and mechatronics experiments based on the NI LabVIEW control and simulation software and NI ELVIS workstation;
- the NI Digital Electronics FPGA Board based on Xilinx Spartan-3E FPGA can be used to teach introductory and advanced digital electronics concepts using either the LabVIEW FPGA module or Xilinix ISE tools.

F. *Student interaction*

The intuitive dataflow programming of LabVIEW, as it has already been presented above, is probably the main reason that students quickly become comfortable with this environment. On the other hand, LabVIEW offers attractive graphical interfaces, which are also very appreciated by students. It contains a large palette of common objects (graphs, charts, knobs, dials, switches, thermometers, Boolean lights, tab controls etc.), whose behavior and appearance can be customized through interactive property pages and which are suitable for the most applications. At the same time, specialized plots and graphs, such as Smith, Bode, Nichols, Nyquist, radar, polar etc., greatly enhance the scientific visualization capability of LabVIEW, allowing students to explore more complex concepts. For instance, within the EMC course provided by our faculty, we use the polar plots in LabVIEW to assist them in better understanding the radiation properties of the antennas (Figure 6).

Another interesting aspect related to data presentation is the LabVIEW ability to display applications on the web. Once a VI has been written, by following just a few simple steps, the LabVIEW built-in web server allows controlling its front panel directly from a web browser. In this way, sophisticated instruments like spectrum or network analyzers, which usually are very expensive pieces of equipment, will become available for all the students in a classroom if they are remotely accessed via LAN. Because of its simplicity, this modality can also be used for complementing the student homework with remote experimentation over the Internet, opening the door for virtual laboratories.

V. CONCLUSIONS

The presented paper has taken into account the authors' experience, results reported in literature by other authors and specific documents offered by software vendors. Thus, based on these data, we can conclude that:

- initially developed for industry, the graphical programming packages and their add-ons are now great tools for teaching and learning electrical engineering concepts within the university courses;
- although the most graphical programming environments are expensive, student versions of these products are available at affordable prices, providing a functionality close to that of the professional software;
- a number of visually-oriented software products, designed to eliminate the programming aspects, are also available. Because of their simplicity, they may serve as alternative or complementary solutions in many educational projects;
- by using LabVIEW, "de facto" standard for graphical programming in measurement, test and control, it is possible to create a large variety of teaching resources. Its intuitive nature and interactive user interfaces are key-features that make both educators and students to include LabVIEW in their applications.

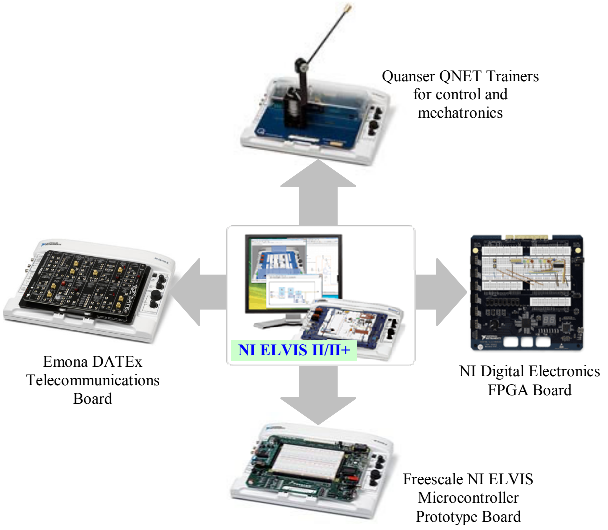
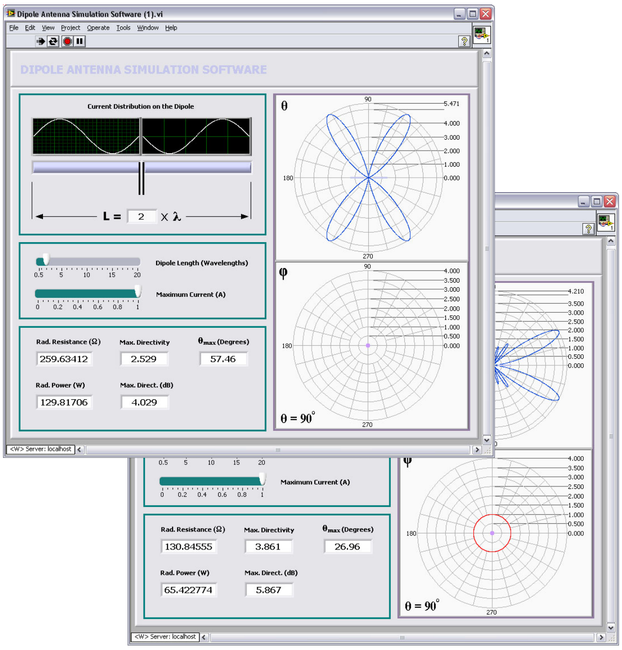

Quanser QNET Trainers for control and mechatronics
Emona DATEx Telecommunications Board
NI ELVIS II/II+
NI Digital Electronics FPGA Board
Freescale NI ELVIS Microcontroller Prototype Board

Figure 5. NI ELVIS II/II+ and plug-in modules for teaching multiple disciplines


Dipole Antenna Simulation Software (1).vi

DIPOLE ANTENNA SIMULATION SOFTWARE

L = 2 x λ

Rad. Resistance (Ω): 259.63412
Max. Directivity: 2.529
θ_{max} (Degrees): 57.46
Rad. Power (W): 129.81706
Max. Direct. (dB): 4.029

Rad. Resistance (Ω): 130.84555
Max. Directivity: 3.861
θ_{max} (Degrees): 26.96
Rad. Power (W): 65.422774
Max. Direct. (dB): 5.867

File Edit View Project Operate Tools Window Help
Current Distribution on the Dipole
Dipole Length (Wavelengths)
Maximum Current (A)
θ
φ
θ = 90°
90
180
270
<W> Server: localhost

Figure 6. Simulation tool for teaching the radiation properties of the dipole antenna of arbitrary length

ACKNOWLEDGMENT

The authors wish to thank Professor Alexandru Salceanu, dean of the Faculty of Electrical Engineering, for helpful discussions regarding this paper and his continuous support.

REFERENCES

[1] M. Parten, "Using Virtual Instruments in a Measurements Laboratory", American Society for Engineering Education Annual Conference & Exposition, 2003.

[2] H. Sumali, "An Instrumentation and Data Acquisition Course at Purdue University", American Society for Engineering Education Annual Conference & Exposition, 2002.

[3] B. Andò, S. Baglio, A. Beninato, S. La Malfa, N. Pitrone, "Advanced Educational Tools in Measurement and Sensors: from remote monitoring systems to magnetic fluids", in *International Journal of Education and Information Technologies*, Vol. 3, No. 1, 2009, pp. 75-84.

[4] C.S. Martis, H.C. Hedesiu, L. Szabó, B. Tataranu, F. Jurca, C. Oprea, "Electrical Machines Virtual Laboratory: Grid Connection of a Synchronous Generator", 12th International Power Electronics and Motion Control Conference, 2006.

[5] Q.H. Arsalan, A. Filbeck, T.W. Gedra, "Application of virtual instrumentation in a power engineering laboratory", 45th Midwest Symposium on Circuits and Systems, 2002.

[6] N. Swain and R. Korrapati, "Design and Development of Virtual Instrument (VI) Modules for an Introductory Digital Logic Course", IJME-INTERTECH International Conference, 2006.

[7] C. Fosalau, L. Nita, M. Antoniu, "Virtual instrument for digital signal processing learning", 3rd International Conference on Electrical and Power Engineering, 2006.

[8] E. Brenner, "Virtual Instruments for the Circuit Analysis with Interactive Parameter Modification", in *International Journal of Online Engineering (iJOE)*, Vol. 2, No. 3, 2006.

[9] K.W.E. Cheng, C.L. Chan, N.C. Cheung, D. Sutanto, "Virtual Laboratory Development for Teaching Power Electronics", IEEE 33rd Annual Power Electronics Specialists Conference, 2002.

[10] J.P. Keller, "Teaching PID and Fuzzy Controllers with LabVIEW", in *International Journal of Engineering Education*, Vol. 16, No. 3, 2000, pp. 202-211.

[11] N.K. Swain, J.A. Anderson, "Study of Electrical Power Systems Using LabVIEW Virtual Instruments (VI) Modules", IAJC-IJME International Conference, 2008.

[12] T.E. Hulbert, R.B. Angus, "Preparing a Virtual Engineering Environment Laboratory Instructional Package", American Society for Engineering Education Annual Conference & Exposition, 2002.

[13] P. Marino, J. Nogueira, H. Hernandez, "Electronics Laboratory Practices Based on Virtual Instrumentation", 29th ASEE/IEEE Frontiers in Education Conference, 1999.

[14] I.G. Pacheco, J.G. Matias, "A Methodology Based on Effective Practices to Develop Educational Software", in *Computacion y Sistemas*, Vol. 11, No. 4, 2008, pp. 313-332.

[15] National Instruments Corp., "DASYLab to LabVIEW Migration Guide", 1999.

[16] Capital Equipment Corp., "TestPoint Quick Start Manual", 2001.

[17] N. Adorno, "Developing a LabVIEW™ Instrument Driver", Application Note 006, National Instruments Corp., 1998.

[18] http://www.ni.com/labview/family.

[19] National Instruments Corp., "NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS)", User Manual, 2003.

AUTHORS

Eduard Lunca is a lecturer with the Electrical Measurements Department, Faculty of Electrical Engineering, Iasi, Romania (elunca@ee.tuiasi.ro).

Silviu Ursache is a PhD student with the Electrical Measurements Department, Faculty of Electrical Engineering, Iasi, Romania (silviu_ursache@ee.tuiasi.ro).

Oana Neacsu is an assistant professor with the Electrical Measurements Department, Faculty of Electrical Engineering, Iasi, Romania (oneacsu@ee.tuiasi.ro).

Manuscript received 24 July 2010. Published as submitted by the authors 20 January 2011..