

## THE VIRTUAL LABORATORY FOR THE DISABLED

Michael Duarte<sup>1</sup> and Brian P. Butz<sup>1</sup>

**Abstract**— *The objective of the Virtual Laboratory for the Disabled is to create a realistic, real-time, electrical engineering virtual laboratory. This project targets individuals who do not have adequate mobility of their upper bodies to perform laboratory experiments. To provide a more realistic and enhanced learning experience, the users of the virtual laboratory are allowed the freedom to build and test a wide variety of realistic electrical circuits, and be able to perform curriculum-based experiments. The main goal is to create an environment similar to a real electrical engineering laboratory, and to offer the user a way to learn the different aspects of instrumentation and circuitry.*

**Index Terms** — *Interactive software, Multimedia, Virtual laboratory*

### INTRODUCTION

According to the Center for Disease Control [1], there are 13.6 million individuals who have limited hand use and another 16.3 million who have mobility limitations. In the field of science and engineering, there are approximately 109,700 persons with motor disabilities employed in the United States [2]. In addition, approximately 31,300 students with motor disabilities were registered in science and engineering programs in 1995 [3]. The Virtual Laboratory for the Disabled (VLD) would help a considerable number of these individuals and encourage others interested to enter the field of electrical and computer engineering. Presently, disabled individuals with motor disabilities have a difficult time with the laboratory portion of the curriculum. At most, individuals with limited or no use of their arms and hands could only watch their lab partners perform the laboratory experiments. This is not good enough for a quality laboratory experience.

The purpose of the Virtual Laboratory for the Disabled is to provide a disabled student with motor disabilities realistic laboratory experience that can be done at the student's pace while providing a good solid curriculum based background in circuit experimentation.

Recent advancements in computer technology and availability have allowed the computer industry to develop hardware and software applications that address the needs of the physically disabled. Circuit simulation software has existed for some time, with the very first simulators being DOS text based programs such as *Pspice*. With the introduction of operating systems with graphical user

interfaces (GUI), better laboratory simulation software became available, such as *Electronics Workbench*. However, these programs contain an interface that is difficult for the physically disabled to use, such as small buttons and an unfriendly breadboard. In addition, because the instruments and components do not look realistic, it can feel like a simulation instead of a laboratory.

The VLD attempts to provide the user with a familiar and realistic environment, using several existing Windows® applications. The main goal is to create an environment similar to a real electrical engineering laboratory, and to offer the user a way to learn the different aspects of instrumentation and circuitry. Furthermore, the laboratory is able to function from a CD-ROM or online for distance learning engineering courses.

### IMPLEMENTATION

This describes the two components of the VLD, the user interface that allows the student to use and control the virtual laboratory, and the communications channel architecture, the framework used by applications packages to communicate with each other.

#### User Interface

The user-interface consists of a breadboard with miniature instruments, with a variety of electrical components (see Figure 1). At this workstation, the user has the freedom to build any type of circuit configuration possible. Typically, the student is given an experiment to complete. The instruments available to the user are a power-supply, a function generator, oscilloscope, and a digital multimeter. The instruments are connected to the holes on the breadboard when the user clicks on the instrument and via-voice enters the coordinate of the hole the user wishes to place a wire. The components, which include, resistors, capacitors, inductors, and jumper wires, connect to the breadboard in a similar way. The coordinates are the letters and numbers seen on the breadboard (see Figure 2). The user enters the letter and number corresponding to the hole on the breadboard where he/she wishes to place a wire or component.

The main goal in the design of the user-interface was to present the user with a realistic environment as well as an environment that a disabled user can manipulate without difficulty. To do this, a particular scheme of design was followed to allow voice recognition software to easily manipulate the environment. For example, a simple voice

<sup>1</sup> Intelligent Systems Application Center, Electrical and Computer Engineering Department, Temple University, Philadelphia, PA 19122

command can move a resistor to a particular point on the breadboard.

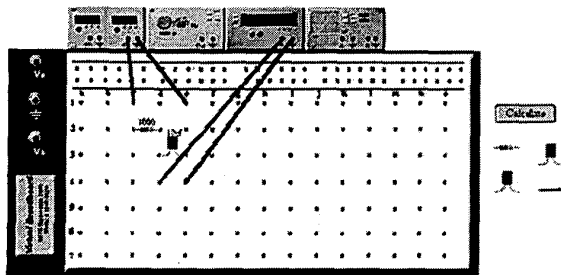


Figure 1. Workstation Window with Circuit

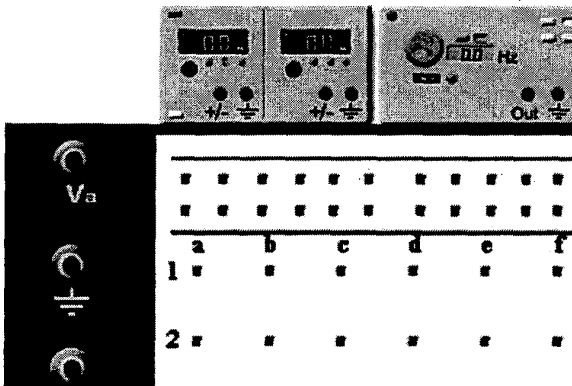


Figure 2. Coordinate View of Breadboard

Each instrument in the workstation has a corresponding larger version that is used to change the parameters of that instrument. With the two input instruments (power-supply and function generator, figure 3 & 4), the user can change the settings and control the type of signal being put through the circuit. The output instruments (multimeter and oscilloscope, figure 5 & 6) can be used to measure the voltage and current of particular parts of the circuit. The user can manipulate the settings on these instruments to view the signal similar to actual instruments.

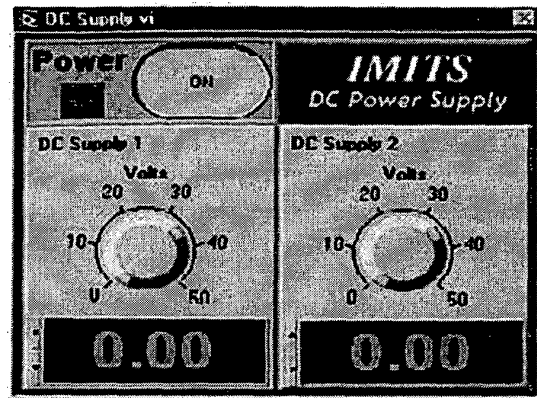


Figure 3. Power-Supply

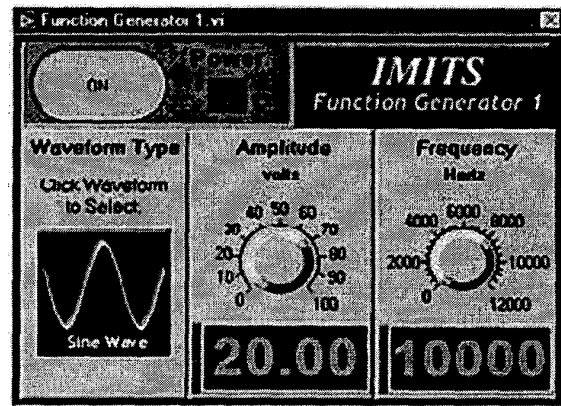


Figure 4. Function Generator

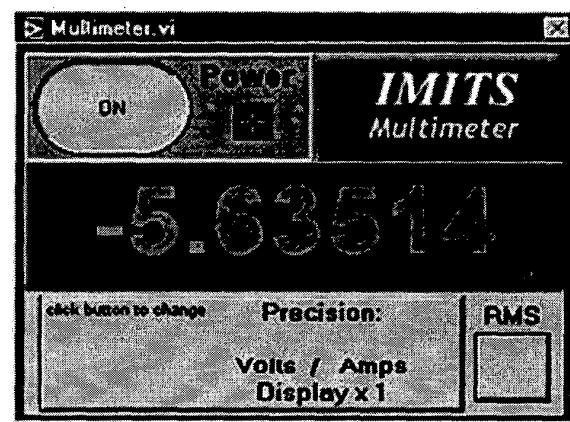


Figure 5. Digital Multimeter

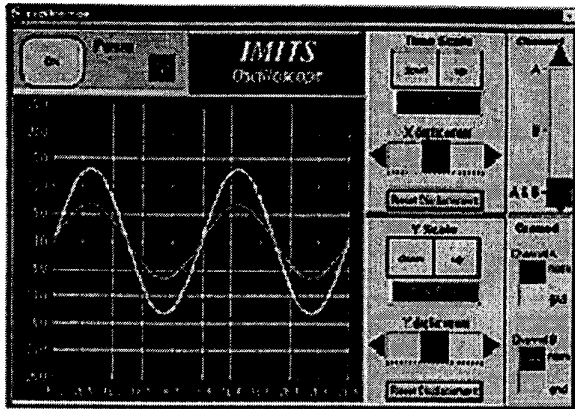


Figure 6. Oscilloscope

**Communications Channel Architecture**

Figure 7 shows the system architecture of the VLD. Creation of the virtual instruments was accomplished through *National Instruments'* software instrumentation package, *LabView*. This software package is typically used in industry and in the laboratory for data acquisition and analysis. Usually, *LabView* is installed and executed from a PC, which is interfaced to several actual laboratory instruments [4]. Its primary function is to take data streams from the real instruments for analysis. However, *LabView* also has the capability of being programmed to customize its interface. Using *LabView's* native "G" programming language, user controlled virtual instruments were created.

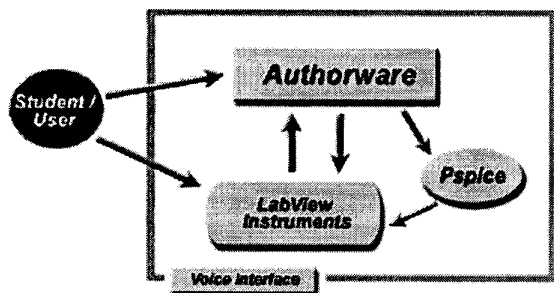


Figure 7. Virtual Laboratory System Architecture

The specific instruments that were modeled are fully adjustable to a range of values found on the equipment in an ordinary electrical engineering laboratory. The digital multimeter is able to measure the current and voltage of a DC circuit as well as measure the RMS value of the current and voltage of an AC circuit. The two-channel oscilloscope is able to display a current or voltage signal of both a DC and AC circuit. The oscilloscope also has the proper buttons and knobs that control the x-y displacement and the x-y ranges. The function generator and the DC power supply can be used as inputs to the circuit. The function

generator supplies a sinusoid or square wave signal, and has an adjustable amplitude and frequency setting. As for the DC power supply; it supplies either a positive or a negative voltage to the circuit.

The virtual breadboard and components were created using *Macromedia's Authorware 5.1*. This is a multimedia-authoring program that allows the user to move objects, type in data, and see their workspace. Here the user can either select components of predetermined values or choose components and enter in values for the components, as well as connect the miniature instruments to the breadboard. After the circuit is built, *Authorware* creates a "netlist," which is a numeric description of the components and how they are connected together. This netlist is sent to *Pspice* for analysis.

*MicroSim's Pspice* is used for creating a mathematical model of the circuit built by the user. *Pspice* is a circuit simulation and modeling package that has been very popular to the engineering community for many years. *Pspice* was chosen because of its ability to accept numeric input as a means for building a circuit. The numeric data that it accepts is the output from the virtual breadboard portion of the project or the "netlist". *Pspice* performs mathematical operations to the netlist and creates an output file containing all the mathematical information of the circuit. This output file is then used by the virtual instruments to display the appropriate information.

The process discussed above needs to work quickly and efficiently, without the user seeing what is happening in order for the virtual laboratory to seem like it is functioning in real-time. Since *Authorware*, *LabView*, and *Pspice* can read and write files, text files are the means of data exchange between the programs (see Figure 8). Through extensive testing of *Authorware*, *LabView*, and *Pspice's* input/output capabilities, it was concluded that using text files is a fast, efficient, and reliable method to create a constant communications channel. This communications channel allows the virtual instruments to update in real-time. The input instruments (function generator and DC power supply) send *Authorware* the settings the user has set through a text file. *Authorware* takes these settings along with the netlist it has created from the virtual breadboard and sends it to *Pspice*. *Pspice* then creates an output file with all the mathematical information of the circuit and sends it to the output instruments (oscilloscope and multimeter) for display. The creation and exchanging of files, as well as the *Pspice* calculation, is hidden from the user (see Figure 8).

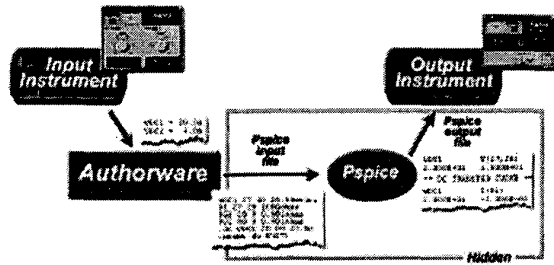


Figure 8. Virtual Laboratory Communication Architecture

The virtual laboratory is made accessible to the physically handicapped via *Dragon Dictate*. *Dragon Dictate* is a speech recognition program that allows a user, via his/her voice, to manipulate the mouse and enter in values for components and instruments. This is the primary input component for the disabled user; however, traditional methods of input (mouse, keyboard) will also work with the virtual laboratory.

Finally, four to six electrical engineering laboratory experiments are incorporated in to the lab. These experiments are presented to the student in a multimedia format using *Authorware*. The user progresses from the presentation of the experiment directly to the virtual lab. These laboratory experiments are a combination of experiments taken from the Instrumentation and Measurement course of the Department of Electrical and Computer Engineering at Temple University.

### FUTURE WORK

A future objective of the virtual laboratory will be to create an intelligent laboratory tutor or assistant that will act like an actual lab assistant. This lab assistant will be able to answer questions, give advice, and figure out what kind of help the user needs within the laboratory experiment. Consequently, the intelligent tutor will have to be able to show what the user is doing wrong and correct or explain the mistakes that he or she has done. With the completion of this objective, a realistic intelligent virtual engineering laboratory would be created.

The intelligent tutor will be created in *CLIPS*. *CLIPS* is a NASA developed expert system shell that can be used to create working expert systems or other intelligent applications. The intelligent tutor will be able to recognize natural speech patterns and decide what kind of assistance the user will need from the tutor. The user will be able to type a question or comment in natural English and the expert system will decipher what the user requires in assistance. Figure 9 illustrates the intelligent tutor system architecture. In addition, the intelligent tutor will monitor the progress of the user and interrupt the user if it decides the user could use some other type of help.

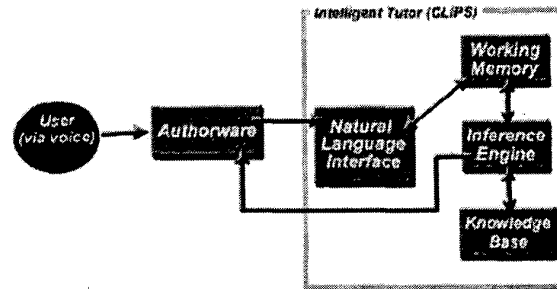


Figure 9. Virtual Laboratory System Architecture

Using *Authorware's* ability to track users movements, the experts system will constantly receive information on what the user is doing in the virtual laboratory. This information will be used as the facts the expert system will need to evaluate the performance of the user. The expert system (intelligent tutor) will have a constant feed of what the user is doing, and will determine if the user knows what he or she is doing, and will attempt to correct the flaws of the user, if he or she is incorrectly performing the steps of the experiment. If a user decides to ask for help by posing a question to the intelligent tutor, the expert system will evaluate the users performance up to that point, and use the question posed to narrow down the users problem, by searching its knowledge base. The question posed by the user will have to be broken down in the form of key words in order to search the knowledge base and determine exactly what the user wants to know. With this knowledge from the user and the knowledge base, the intelligent tutor will then give the appropriate tutoring or assistance to the user.

### ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support provided for this work by the National Science Foundation's Division of Undergraduate Education through grant DUE #9952291.

### REFERENCES

- [1] Center for Disease Control (1995). National Health Interview Survey, 1994.
- [2] National Science Foundation (1997), Award Abstract #9710548, *SBIR Phase II: Computer Simulation of Science and Technical Laboratory Exercises for Physically-Disabled Students*, <http://www.fastlane.nsf.gov/servlet/showaward?award=9710548>.
- [3] National Center for Education Statistics. The 1996 National Postsecondary Student Aid Study data system, 1996.
- [4] Wells, Lisa K., Travis, Jeffrey, *LabView: for Everyone*, Prentice Hall PTR, New Jersey, 1997.  
et al.