

# A LabVIEW-Based, Interactive Virtual Laboratory for Electronic Engineering Education\*

ZHOU YI, JIANG JIAN-JUN and FAN SHAO-CHUN

Department of Electronic Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, P.R. China. E-mail: longhairbrother@hotmail.com

*Virtual laboratories extend the application area of virtual instrumentation to the web, which leads to an openly integrated environment for sharing teaching and learning materials and hardware and software resources. An interactive Virtual Laboratory for Electronic Engineering Education (VLE) was implemented based on LabVIEW and other engineering education software such as MATLAB Simulink, MS Office and SQL Server. It consists of two sections: virtual simulation and real-time measurement. The system architecture, hardware configuration and system implementation are presented. Compared with existing virtual laboratories, VLE provides significant advantages and improvements in development efficiency, system expandability, flexibility for working independently, and multithreading safety.*

## INTRODUCTION

VIRTUAL LABORATORY is a virtual experiment environment in which users can operate a series of graphical units, each one representing an experiment object, to conduct remote virtual or veritable experiments via the internet by local operation of the mouse or keyboard.

With the help of Virtual Laboratory students are able to conduct online experiments 24 hours a day, seven days a week, without limitation of space and time. Moreover, Virtual Laboratory makes it possible for students to do experiments selectively, which gives them a new way to put more energy into the fields they are interested in. This flexibility enhances the student's enthusiasm for learning and improves the interaction between teaching and learning as well.

The following existing virtual laboratories, implemented by different methods, are good paradigms for student experiments.

A Virtual Engineering/Science Laboratory [1] at Johns Hopkins University makes it possible to simulate engineering and science laboratory projects on a remote computer. With internet access, it offers students 'virtual laboratories' via the world wide web. The experiments are written in JAVA, which requires students to access them using a web browser. Students first download a JAVA applet to employ as a GUI, which also builds up the communication between client and server; they then construct their experiment system by using the objects on the GUI; after defining the experiment's working conditions and submitting

it, the simulated experimental results are returned to them via the server. This Virtual Laboratory is designed to introduce students to experimentation, problem-solving, data-gathering, and scientific interpretation early on in their careers.

Carnegie Mellon's Virtual Lab [2] allows undergraduate students to conduct actual experiments at any time and from any location by remote computer access to laboratory instruments. The instrumentation available in the core laboratory consists of a function generator, a digital multimeter, and a digital oscilloscope. The instruments are controlled with HP-VEE software running by means of Windows on a personal computer. Remote control of the computer is made possible using Timbuktu software, and PC/TCP software provides the link to the internet. Live video is available using a Connectix camera and QuickPICT software. This technology greatly enhances the flexibility of laboratory education and introduces students to remote experimentation.

Engineering Laboratories on the Web [3] at the University of Tennessee at Chattanooga is composed of a server machine and five client machines, each controlling a different physical system. Each of the client computers communicates with the server via the NT file server and client software written with LabVIEW. The clients scan for command files on the server, which are produced by the LabVIEW web server each time an experiment is run by the user. The client software reads the command files off the server and decodes them into variables used by the Data Acquisition (DAQ) cards to control the physical system. The hardware of the physical system is monitored and controlled using National Instruments' DAQ cards on the client computers. The

\* Accepted 7 November 2004.

DAQ cards allow the software to communicate with the outside world using low-level analog and digital signals.

In the examples described above, the virtual laboratory is mostly built on client/server architecture, which is generally one of the following three types:

1. *Software sharing virtual laboratory.* The server shares local simulation software by processing experiment commands sent from clients and returning experiment results back to them after accepting experiment requests. This system does not deal with actual experiment hardware; instead, software on the server simulates the experiment procedures and furnishes the client with simulated experiment results, as with Virtual Engineering/Science Laboratory.
2. *Instrument sharing virtual laboratory.* The server also accepts experiment requests and commands from clients. However, instead of sharing software the server uses these commands to control instruments connected to it and makes them complete the actual experiments. After that, the server returns the experiment results to the client to achieve hardware and data sharing, as with Carnegie Mellon's Virtual Lab.
3. *Remote control virtual laboratory.* As with the instrument sharing virtual laboratory, users can control each experiment procedure, rather than just start and finish an experiment. This requires much stricter real-time communication between the server and clients, as with Engineering Laboratories on the Web.

An interactive Virtual Laboratory for Electronic Engineering Education (VLE) was constructed to implement the experiment courses for the Department of Electronic Science and Technology at Huazhong University of Science and Technology (HUST). It also conveniently and efficiently overcame the problem of shortage of training resources.

The trial group was made up of ten undergraduates, all LabVIEW beginners, and two postgraduates with LabVIEW programming experience. Up to 40 independent experiments were completed in one summer vacation. VLE

has been used as the experiment platform for specialty courses in electronics for undergraduates for one semester, which works stably with high-density usage and provides free access to students in every department any time of the day or night.

## SYSTEM ARCHITECTURE

The studied VLE mainly consists of two sections: virtual simulation and real-time measurement, as shown in Fig. 1.

The *virtual simulation* section mainly focuses on theory demonstration experiments, and runs without veritable experiment instruments. Student users are able to access it with a web browser and operate downloaded GUIs embedded in web pages to conduct virtual experiments. Simulation application programs on the server process users' commands and return the simulated results to them to complete the virtual experiment.

The *real-time measurement* section mainly focuses on instrument-sharing experiments and remote control experiments. The detailed experiment procedure is as follows. First, the student user needs to download a Multimedia Assistant module, which provides an experiment GUI showing the operations to be conducted. Then the student needs to operate this GUI to pass a pre-test and become acquainted with the experiment contents and procedures. After that, a real-time measurement module, the core component, is downloaded to implement interaction and communication between the server and clients and to complete the real-time measurement.

## HARDWARE ARCHITECTURE

VLE is based on Browser/Server/Database/Application (BSDA) architecture which consists of four modules: client, web server, application server and instrument, as shown in Fig. 2.

The web server is used to provide world wide web service and to generate the virtual experiment platform embedded in a web page; the application server is used for data acquisition, data processing,

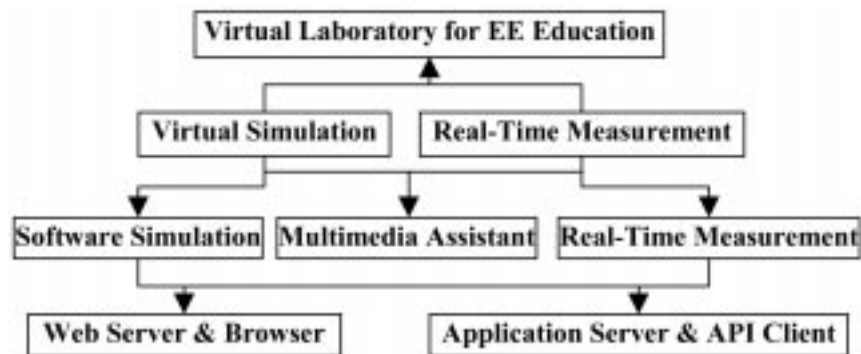


Fig. 1. System architecture of VLE.

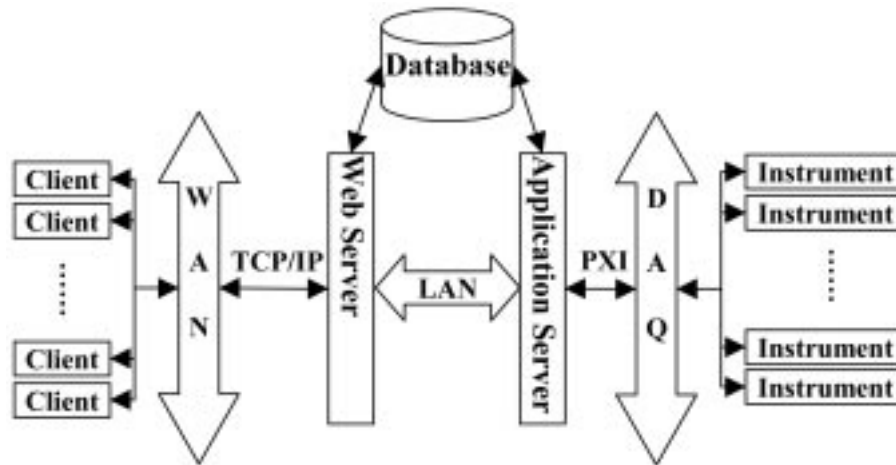


Fig. 2. Hardware architecture of VLE.

and instrument control; and the database is used to store user information, experimental data and to support ASP service.

The web server is constructed on a Dell Power-Edge tower server. It is embedded with two Intel Xeon processors at 2.8GHz, 2GB ECC DDR RAM, three 36GB SCSI RAID, three 300w redundant power supplies and one Broadcom gigabit network interface card, which makes it meet the requirements of high performance calculation, multithreading safety, and broadband data transmission.

The application server works on the NI-PXI hardware series. A PXI-1000B chassis embedded with multiple PXI DAQ cards serves as the application server. PXI-8176 is a high-performance Pentium III-embedded controller for modular instrumentation use; PXI-6070E and PXI-6071E multifunction DAQ cards deliver high-performance, reliable data acquisition capabilities to meet a wide range of application requirements; PXI-5102 is a dual-channel digitizer used to make high-speed time and frequency domain measurements; PXI-2501 performs as a multiplexer or matrix switch to route signals from various kinds of instruments to various test points so as to achieve automatic jumper wiring; the PXI-1422 IMAQ card performs high-speed and high-resolution digital image capture to meet the requirements of PCB tests, IC chipset tests and circuit checking; PXI-7344 is a high-performance servo and stepper motion controller for dealing with motion control systems.

A SCXI-1000 chassis embedded with SCXI-1320 and SCXI-1125 signal conditioning cards is used to amplify low-level current and voltage signals, and to filter noise so as to improve measurement accuracy.

The new NI-Educational Laboratory Virtual Instrumentation Suite (ELVIS) consists of LabVIEW-based virtual instruments, a multifunction data acquisition device, a custom-designed bench-top workstation and a prototyping board,

which is used to construct prototypes of circuit design, motion control and DAQ systems.

## EXECUTION PROCEDURE

VLE is performed in three steps [4, 5]: pre-experiment, experiment and post-experiment. The pre-experiment step is mainly for obtaining the necessary background information on the project and downloading GUIs for the experiment platform. The experiment is interactively executed by operating the virtual instrument platform to collect data and display the processed results. The post-experiment step finishes with a Q&A test and an experiment report about the project. Figure 3 shows the system's working procedure.

Before each experiment is conducted, the student user needs to pass a pre-test regarding the experiment content and procedure. One pre-test platform is shown in Fig. 4. The student can observe changes in the virtual plot at the top left of the screen by pushing buttons and adjusting knobs to vary the control values on the instrument panel. After that, he or she is allowed to operate the veritable real-time experiment to measure the DC characteristics of a transistor in a remote laboratory by varying the control parameters in the experiment GUI platform, thus obtaining the  $I_c - V_{ce}$  curve.

## SYSTEM IMPLEMENTATION

Each experiment is implemented in two editions. One is based on the Web Server & Browser architecture; the other is based on the Application Server & API Client architecture. With the former architecture, only one single user can have full control of the remote experiment and others can only view the experiment results until the experiment control is released. With the latter architecture, a FIFO queue is constructed to handle

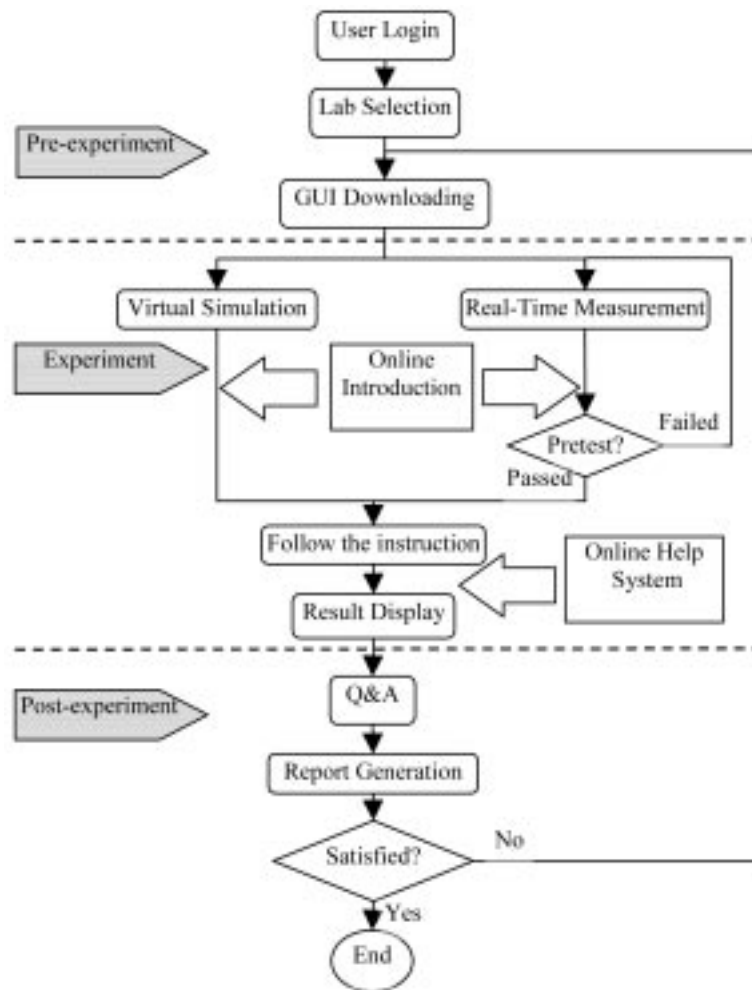


Fig. 3. Working procedure of VLE.

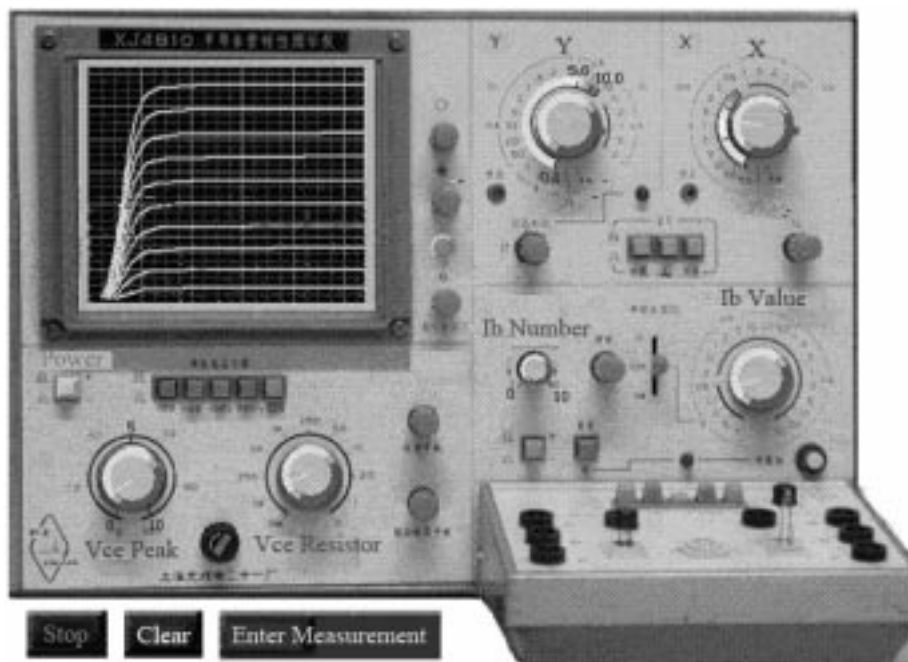


Fig. 4. Pre-test platform of Transistor DC Characteristic Measurement experiment.

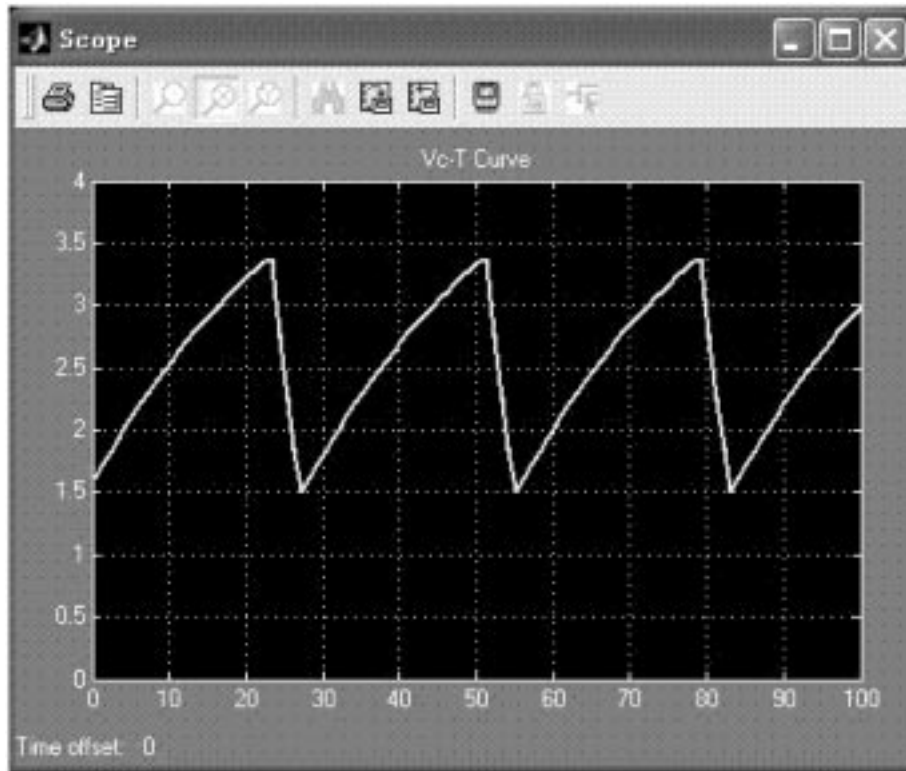


Fig. 5.  $V_c$ - $T$  curve of Multivibrator Circuit in MATLAB Simulink.

multiple users' command requests. The server completes each command as soon as possible, and then proceeds with the next one, which makes the multiple users feel that they are controlling the remote experiment synchronistically.

#### *Web server and browser architecture*

This architecture is based on the built-in Web Publishing Tool [6] in LabVIEW, which makes use of the LabVIEW web server to create HTML documents and install front panel images and VI applications in a web page.

The development procedure is as follows. First, a stand-alone application program is developed with LabVIEW programming. After running a test, it is embedded into a web page and automatically gets a unique URL by using the Web Publishing Tool. Next, the student inputs the upper URL and controls a web browser to access the published front panel and conduct virtual experiments. After the experimental data is processed, the LabVIEW web server, implementing the communication between the user and the server, sends the experiment results on a web page back to the student user to complete the experiment. Meanwhile, the Remote Panel Connection Manager embedded in LabVIEW is used to monitor online users and balance network traffic.

The simulation algorithm section is implemented with Matlab Simulink. The Simulation Interface Toolkit [7] is installed to create a LabVIEW

user interface for a Simulink model, to convert a Simulink model into a Dynamic Link Library (DLL), and to run the simulation model on the web server. In this way, the student can manipulate the model parameters and view the output data of the Simulink model in LabVIEW.

Figure 5 shows the  $V_c$ - $T$  curve of the Multivibrator Circuit experiment simulated in MATLAB Simulink, as imported into the front panel of the Multivibrator Circuit VI application. Next, the VI application is published as a web page, as shown in Fig. 6. Thus student users can access the VLE to conduct experiments just by means of a web browser.

In the upper Virtual Simulation experiments, the web server is used to share simulation software without instrument hardware; however, in the ensuing real-time measurement it is connected with experiment instruments through DAQ and SCXI hardware to accomplish data transmission between clients and remote instruments.

Figure 7 shows the front panel of an Inflection Signal Generator real-time experiment. The stand-alone VI application is also published as a web page. Thus the student user can access the VLE to operate this platform in the web page by changing the circuit parameters and controlling a speaker to perform music in the remote laboratory.

In conclusion, the Web Server & Browser architecture suits the thin client system. The client just sends experimental commands to and receives experimental results from the web server instead

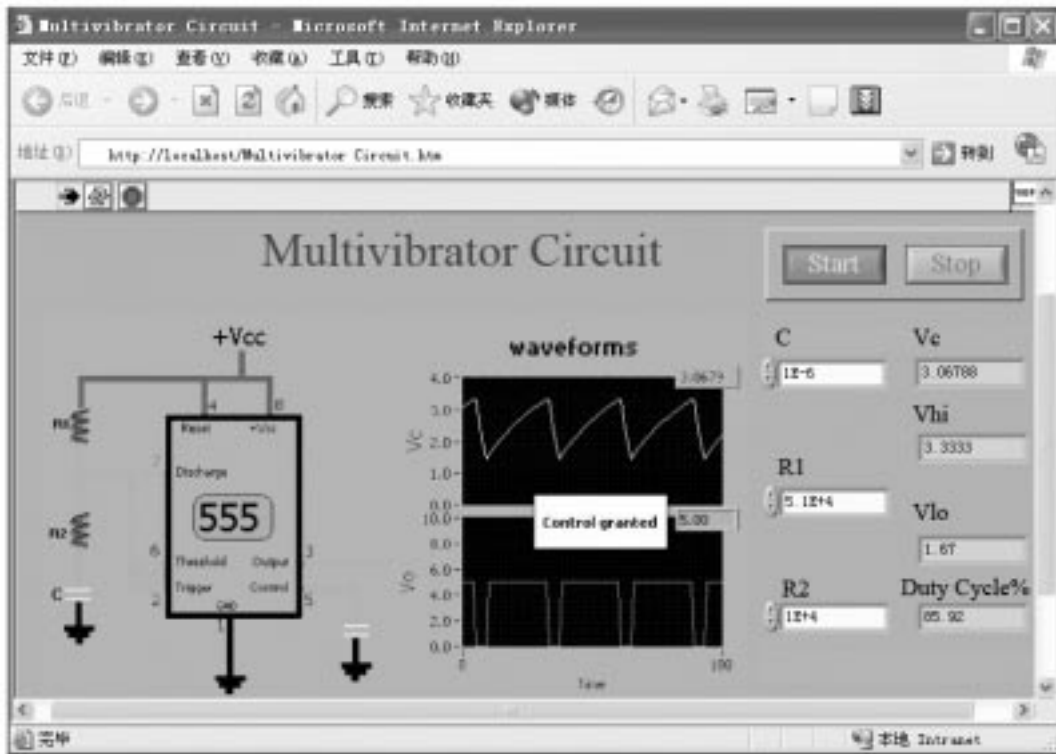


Fig. 6. Front panel of Multivibrator Circuit VI application embedded in a web page.

of processing the experiment data itself. Therefore, only a web browser is needed by the client to conduct this kind of experiment. However, the web server has to have powerful calculation capability, wide transmission bandwidth and high multithreading safety.

*Application server and api client architecture*

This architecture is based on LabVIEW programming, especially the built-in TCP/IP module which carries out the communication between the Application Server application and the API Client application written in LabVIEW.

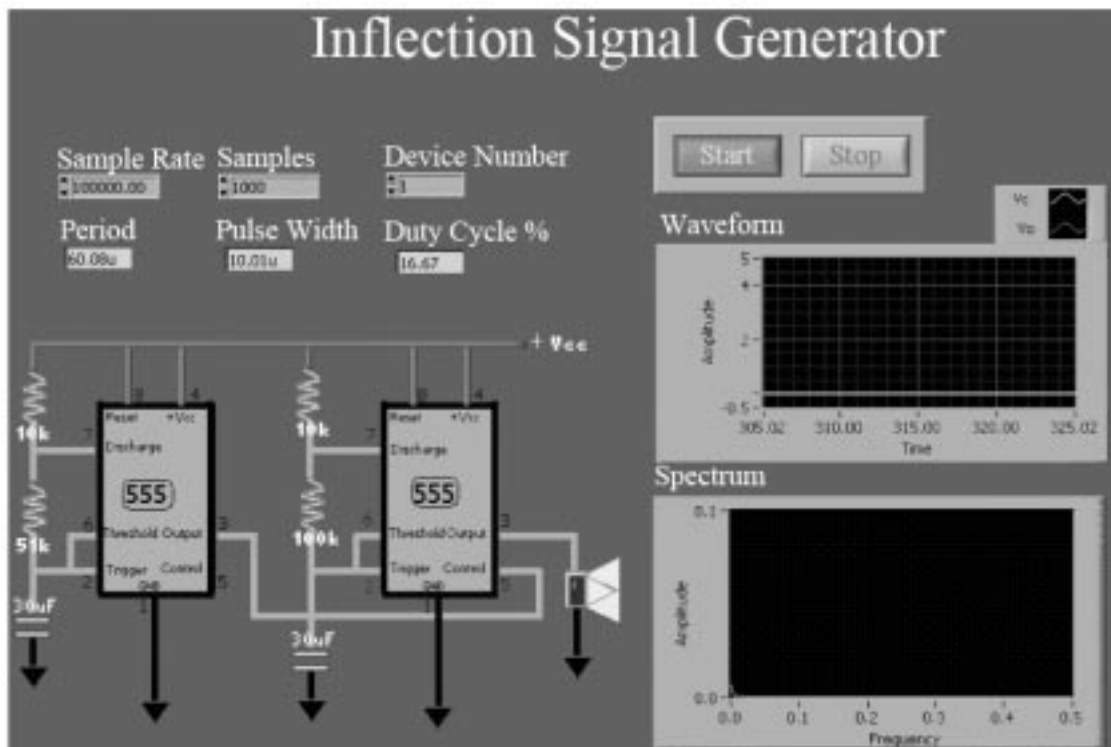


Fig. 7. Front panel of Inflection Signal Generator experiment.

The Application Server application is used for DAQ, data transmission, user management and network traffic balance. The API Client application is used for GUI platform provision, receiving experiment data and displaying experiment results.

In this architecture the API Client first builds a connection with the Application Server; next, it sends control commands to and receives experiment data from the Application Server interactively; it then processes the experiment data by itself; finally, it displays the results in various forms.

The workflow is as follows. The API Client first transmits user information to the Application Server; after ID verification, the Application Server accepts experiment commands from the API Client to conduct the experiment and sends experiment data packets back; later on, the API Client itself processes these data packets, displays the experiment results and generates an experiment report; finally, the Application Server disconnects from communication with the API Client after the experiment is completed and responds to other API clients' requests.

An Experiment Management application is constructed to display the experiment plots of real-time data flow and the online user's information, such as user name, user number, IP address, experiment name and so on. With the help of this application, the administrator can monitor the real-time state of each experiment and exclude any illegal user.

In conclusion, the Application Server & API Client architecture suits the fat client system. For experiment data to be processed by the API Client, high performance hardware is required for the API Client as well as wide transmission bandwidth and running stability.

#### Database

Each experiment application shares the same database sub-VI, which contains two modules: 'Insert into Database' and 'Select from Database'. The former is built into all real-time applications to store user information and experiment data on a database, which is implemented by the LabVIEW Database Connectivity Toolset [8]. This toolset is an add-on tool package for accessing databases. It contains a set of high-level functions for performing the most common database tasks and a set of advanced functions for customized tasks.

Experiment data to be stored in a database is first bundled into a uniform format and then sent to the 'Insert into Database' sub-VI, which creates a connection between the application and database by a Universal Data Link (UDL) and automatically inserts experiment data into the designated tables. Meanwhile, when the student user needs to make use of the data stored in the database, he or she can control the 'Select from Database' module to fetch data from the database and process it in different ways.

MS Access XP and SQL Server 2000 were

installed on the Application Server, and relative ODBC drivers were installed and appropriately configured. Therefore any user can use a UDL to automatically choose one database to read and store experiment data, on condition that the user name and password are correct.

#### Online help and report generation

To improve the vitality of VLE, flash animations introduce the online help system. Each experiment has a Multimedia Assistant, from which the student user can get a series of basics and background information for the experiment. A flash animation is first completed in the Macromedia FlashMX environment. After that, it is embedded into a web page. In this way the student can access the online help system just by means of a web browser.

When an experiment is finished, an experiment report embedded with the experiment data, result charts, or graphics is generated in the form of a web page, which is achieved by the help of the Report Generation Toolkit [9]. It provides a set of flexible easy-to-use VIs to generate an experiment report using MS Word and Excel.

## CONCLUSION

The LabVIEW-based, interactive VLE now consists of four classes: DSP, circuit experiment, magnetic measurement and electronic component measurement. Compared with existing virtual laboratories, it provides the following advantages and improvements:

1. *High efficiency.* The development team was mainly made up of ten LabVIEW-beginner undergraduates. With the help of the easy-to-use G Language programming tool LabVIEW, it took only four weeks to finish the main experiment programming section of 40 experiment applications, which would have been quite difficult to achieve with other programming platforms.
2. *High expandability.* The abundant instrument library embedded in LabVIEW contains more than 2,200 instrument drivers from more than 150 vendors. What is more, it can also interact with application software (such as MS Office), C language, Windows API, Matlab and HiQ using external interfaces such as DDE, CIN, DLL, etc. As a result, it is more convenient to extend VLE with third-part applications and instruments such as Agilent network analyzer, Fluke multimeter, and so on.
3. *Independency and flexibility.* The Web Server & Browser architecture was developed based on HTML, which was independent of the operating system of the client. Only a web browser is required by the client to access VLE and conduct experiments. Since the client just sends commands to and receives results from the

web server, this architecture suits software-sharing experiments and instrument-sharing experiments in the thin-client system.

The Application Server & API Client architecture was developed based on LabVIEW programming. However, by using the NI Application Builder tool, LabVIEW-dependent VI applications can be saved as LabVIEW-independent executable applications to run without LabVIEW. Since the experimental data downloaded from the Application Server is processed by the client, this architecture suits instrument-sharing experiments and remote-control experiments in the fat-client system.

4. *Multithreading safety.* Each user of VLE can conduct experiments without disturbing others. In the Web Server & Browser architecture, control of the experiment is held by just one user and others can only view the experiment results. However, each user is required to complete the experiment in a limited time in order to share the control of the whole system. In the Application Server & API Client architecture, a FIFO queue is constructed to handle multiple-users' command requests one after another in a short time, which makes the multiple users feel that they are controlling the remote experiment synchronistically.

VLE was demonstrated successfully in the branch school about twenty miles away from the central laboratory. Virtual simulation experiments and real-time measurement experiments were conducted stably for several hours while attached to live video stream in multithreading. What is more, it has served as the experiment platform for specialty courses in electronics for undergraduates for one semester with free access to students in every department at any time of the day or night.

Feedback from student users is encouraging:

- First, it is time-saving, considering that each experiment takes just five or ten minutes instead of half a day, as before, most of which was wasted in waiting around or making simple

repetitive adjustments and observations. Some student users even completed three or more experiments in one class. This gives them an effective way of focusing more on experiment theory and knowledge expansion.

- Second, the Multimedia Assistant, pre-test and Q&A greatly increase student users' interest in what they are doing and provide them with a lot of information about it. This multimedia teaching and learning process is more stimulating for students, which helps to fix what they have learned more securely in their minds. At the same time, as the experiment is run on the web, students are able to make good use of internet resources to get what they need immediately, which greatly enriches their knowledge of the field of science.
- Moreover, students are able to use existing virtual experiment objects to construct their own experiment system and obtain the experiment results to check it, which so many students find quite difficult using traditional experiment instruments. But if students are unclear about any experiment they have conducted on the web, they can still go to the lab to see the real experiment process, and even to conduct it if they so wish. However, hardly any students do this and nearly all of them get good marks in their tests.

The lab faculties' responses are inspiring, too. VLE makes it convenient to check every student's work online by monitoring the real-time data flow and searching for student information in the database. If some students encounter a problem, they can communicate with each other or ask the tutor for help by means of chat tools such as MSN messenger. Meanwhile, the experiment report is sent in by e-mail in a uniform format, and the teacher can review it online.

*Acknowledgements*—This work was supported by the National Education Reform Project of New Century. The authors also acknowledge the donation of software and technical aids by the National Instruments Corporation.

## REFERENCES

1. A Virtual Engineering/Science Laboratory (<http://www.jhu.edu/~virtlab/virtlab.html>).
2. Carnegie Mellon's Virtual Lab (<http://www.ece.cmu.edu/~stancil/virtual-lab/application.html>).
3. Engineering Laboratories on the Web ([http://chem.engr.utc.edu/Webres/how\\_it\\_works.html](http://chem.engr.utc.edu/Webres/how_it_works.html)).
4. Dongil Shin, En Sup Yoon, Kyung Yong Lee, *et al.*, A web-based, interactive virtual laboratory system for unit operations and process systems engineering education: Issues, design and implementation, *Computers and Chemical Engineering*, **26** (2002), pp. 319–330.
5. R. M. Parkin, C. A. Czarnecki, R. Safaric, *et al.*, A PID servo control system experiment conducted remotely via the internet, *Mechatronics*, **12** (2002), pp. 833–843.
6. National Instruments, *LabVIEW User Manual*, Austin, Texas (2003), pp. 8–10.
7. National Instruments, *LabVIEW Simulation Interface Toolkit User Guide*, Austin, Texas (2003), p. 4.
8. National Instruments, *LabVIEW Database Connectivity Toolset*, Austin, Texas (2001), p. 1.
9. National Instruments, *Report Generation Toolkit for Microsoft Office User Manual*, Austin, Texas (2001), p. 1.



**Zhou Yi** is a postgraduate of the Electronic Engineering Department at Huazhong University of Science and Technology (HUST). He received his B.Sc. degree from HUST and was recommended to further his study as a postgraduate. He currently conducts research on virtual laboratories, distributed measurement systems and virtual instrumentation applications.

**Jiang Jian-Jun** received his Ph.D. degree in Materials Science and Engineering from Zhejiang University in 1995. Currently, he is a Professor and Associate Chair for Undergraduate Studies in the Electronic Engineering Department at HUST. From 1997 to 1998 he worked in the Department of Chemistry at the Technion-Israeli Institute of Technology. From 1999 to 2000 he worked in the Materials Process and Powder Metallurgy lab at the Helsinki University of Technology. He has published over 40 conference and journal papers in the areas of virtual laboratories, GHz microwave permeability and hydrogen storage alloys. His current research interests focus on virtual laboratories and material design.

**Fan Shao-Chun** is a postgraduate of the Electronic Engineering Department at HUST. He received his B.Sc. degree from HUST and was recommended to pursue a Ph.D. degree. He currently conducts research on virtual laboratories and material design.