

AC 2008-1098: DEVELOPMENT OF AN OPERATIONAL AMPLIFIER VIRTUAL LABORATORY BASED ON ILAB ARCHITECTURE AND NI ELVIS

Kayode P. Ayodele, Obafemi Awolowo University

Lawrence O. Kehinde, Texas Southern University, Houston, Texas

Olutola Jonah, Obafemi Awolowo University

Olusoji Ilori, Obafemi Awolowo University

E.O.B. Ajayi, Obafemi Awolowo University, Nigeria

Oladipo O. Osasona, Obafemi Awolowo University, Ile-Ife, Nigeria

Development of an Operational Amplifier Virtual Laboratory based on iLab Architecture and NI ELVIS

Abstract

We have developed and deployed 'OpAmp Lab', a pedagogic operational amplifier virtual laboratory based on the MIT iLab architecture. The architecture is a three-tier architecture that abstracts the various parts of a remote laboratory system into different functional domains. This architecture makes the new virtual Lab experiments scalable and reduces the development and deployment time of new experiments through component reuse.

OpAmp Lab uses the National Instrument ELVIS hardware platform for experimental set-up. Basic Op-amp circuits are realized through reconfigurable Dozen Impedance Op-amp configuration mounted on a multimode switch array. The front end or Client is implemented as a Winform Client, written in C# and uses connectible nodes as a metaphor for back-end hardware configuration. Strong emphasis is placed on the realism of the user interface, as we argue, the user interface is the most important determinant of the quality of a user's educational experience.

We discuss the various components of OpAmp Lab and the lessons learnt in the development of the system as well as the development of pedagogical experiments developed for it. OpAmp Lab has been in use at Obafemi Awolowo University, NIGERIA for the last two years. Faculty and students response has been strongly positive. We provide data showing that the Lab offers a viable means of augmenting the laboratory component of electrical engineering curricula in developing countries.

Keywords: *OpAmp Lab, Virtual Laboratory, iLab*

I. Introduction

Virtual Laboratories (VLs) are spatially or temporally dispersed systems that allow students to carry out activities aimed at achieving the same goals as traditional laboratory experimentation. When utilized for Science and Engineering education, VLs allow students to have round-the-clock access to remote laboratory equipment to realize the experimentation component of their curricula. The advantages of VL used in this way have been noted in the past but arguably the most important of these to institutions in the Third World is the fact that VLs are scalable¹, allowing a single set of equipments to be used by multiple institutions and laboratories to be scaled easily to handle an increasing number of users. This scalability is of critical importance in Nigeria, where severe underfunding of education means that even the most basic equipment could sometimes be beyond the budgets of educational institutions. VLs then offer a viable way for an institution to expose its students to far more laboratory experiment by sharing resources and equipments of other local or foreign institutions.

The role that VLs can play in addressing the weak experimentation component of education in developing countries has only recently become a research problem of interest. Consequently,

very few VL architectures are designed or optimized for the low bandwidth environments that characterize educational institutions in countries like Nigeria. One of the few VL architectures that are suited to such low-bandwidth scenarios is the Batched version of the MIT iLab Architecture^{1, 2, 3, 4, 5}. The iLab Batched architecture is referred to as “iLab-B” in this paper to distinguish it from the iLab Interactive architecture. The iLab-B architecture separates the various parts of a VL in such a way that only a very small amount of data needed to configure the remote system under test needs to go through the low-bandwidth uplink, while all other data lab-related activities are done locally. As a result, the MIT iLab-B architecture was an ideal platform upon which to base the development of a new Operational Amplifier virtual laboratory (“OpAmp Lab”) hosted at Obafemi Awolowo University, Nigeria.

II. Architecture Overview

OpAmp Lab employs the three-tier MIT iLab-B architecture. This architecture abstracts the various functional parts of a VL into three distinct domains^{2, 3, 4}: the Client, the Server and the Service Broker (Figure 1). While the Client and Service Broker tiers are software-only tiers, the Server has software components as well as the system under test, which is usually hardware. The three tiers communicate through Simple Object Access Protocol (SOAP) messages using platform-agnostic XML web services⁵. The use of SOAP allows the message itself to be abstracted from the underlying implementation technology of the various tiers. This is significant since it means that there are no artificial restrictions on the user’s hardware or software configuration or the implementation technology of the two other tiers. For example, any of the tiers could be re-written in any programming language of the developer’s choice and as long as that language can implement and consume web services, the new version would be a drop-in replacement for the old one.

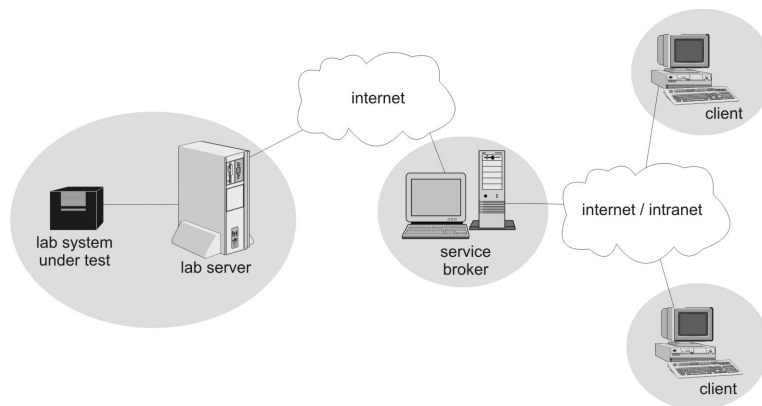


Figure 1: MIT iLab-B Architecture

The Client tier consists of a user interface through which a user interacts with the backend system under test. The user interface is expected to present a metaphor by which a student would relate changes made to interface elements to appropriate configuration changes to the remote system under test. To ensure ease of use, the Client user interface is implemented as a browser-based control, since most browsers would already have the requisite plug-ins and even when they do not, installation is usually easy. To ensure wide compatibility, the controls are developed using a platform-independent programming platform. To date, the most common platform for Client control implementation is Java.

The Service Broker tier is usually located on the user’s computer network. It performs a number of tasks. First, it handles administrative tasks like creation and storing of students’ accounts. To use the lab, a student would need to log into the Service Broker. All session information related to a student’s lab use are tracked by the Service Broker and when the student configures the backend, the description of his configuration, referred to as Lab Configuration are stored on the Service Broker for future access. The second role of the Service Broker is to act as a trusted proxy for the Client for communications with the Server. Using a one-to-many mapping, it is possible for the Server to deal indirectly with many clients through a trusted Service Broker.

The Server tier consists of the system under test, interfacing hardware, and software components residing on the computer to which the system under test is coupled. When a user configures the Client and requests for an experiment to be run, a message called the experiment Specification is passed to the Server, where it gets added to a queue. A software component called the experiment execution engine runs through the queue, parses the experiment specifications, and using the interfacing hardware, configures the system under test appropriately. Signals are then passed into the system and the outputs recorded and returned to the Client through the Service Broker.

III. OpAmp Lab Hardware Components

The OpAmp Lab circuit under test is based on the Dozen-Impedance operational amplifier configuration as reported by Kehinde ⁶. The dozen impedance configuration is an operational amplifier circuit with a dozen impedances connected to it through switches in such a way that various basic operational amplifier circuits are implemented (Figure 2). For example, by closing switches s_2 , s_3 , s_{12} and s_{15} , and substituting appropriate capacitors for impedances Z_2 and Z_3 , the circuit becomes an Integrator or a Differentiator, depending on the natures of the impedances. An LM 324N low power quad operational amplifier was used in implementing the OpAmp Lab Dozen Impedance circuit.

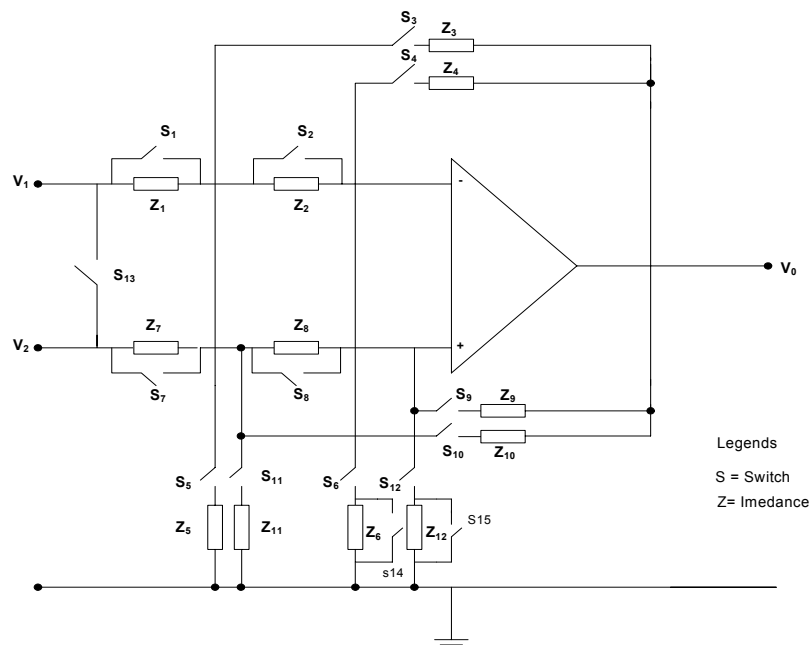


Figure 2: The Dozen-Impedance Circuit

The OpAmp Lab switches were implemented using a multimode switch matrix developed around a PIC18F452 microcontroller and six MAX4664 FET switches (Figure 3). The MAX4664 has ON impedance of 5 Ohms and an OFF leakage current of 5nA. Although the ON impedance is relatively low, it could still cause significant errors in resistance values in the various arms of the operational amplifier circuit. For this reason, the Differential Amplifier configuration of the Dozen Impedance circuit was not used. The switch matrix ran a simple firmware that allowed switch configuration information to be passed to it using the RS232 port of any sufficiently intelligent device. In the OpAmp Lab, the switch matrix was connected to the RS232 port of the Server through a MAX232 level converter.

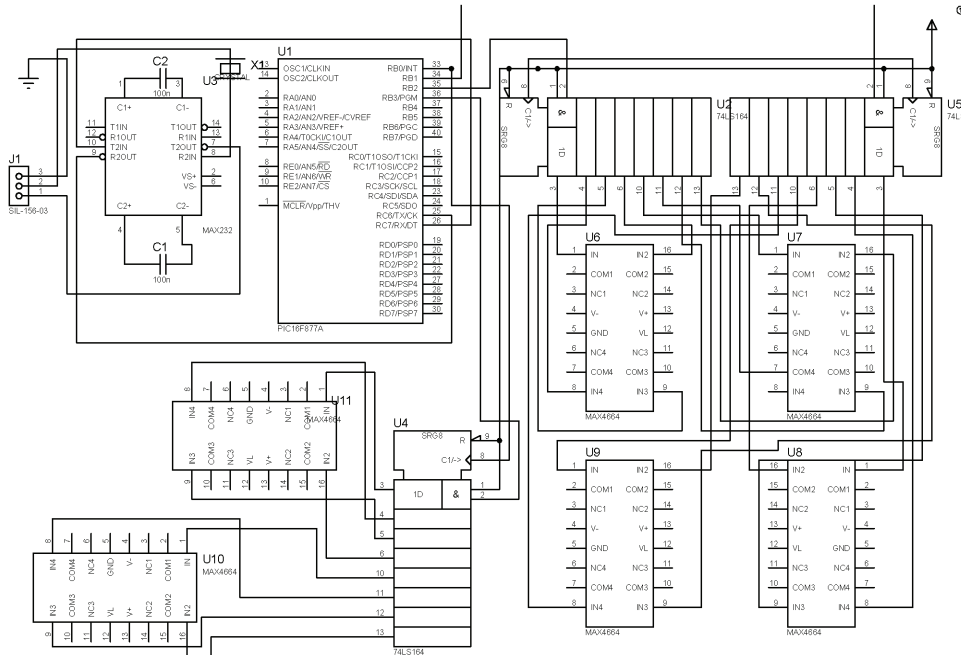


Figure 3: MAX4664 and PIC18F452-based Switch Matrix

The circuit under test was mounted on the prototyping board of a National Instrument (NI) Educational laboratory Virtual Instruments Suite (ELVIS)⁷. NI ELVIS is an instrumentation platform that integrates a set of measuring instruments with a signal generator and a power supply. Interfacing with a computer is achieved through an NI Data Acquisition (DAQ) card. ELVIS instruments typically have bandwidths limited by the capability of the DAQ card, up to a maximum of 500kpsps. The advantage of using ELVIS is that it presents all the instruments needed for instrumentation and measurement needs for a circuit under test in one single package that can be controlled programmatically from a personal computer. For OpAmp Lab, ELVIS v1.0 was connected to the Server through an NI PCI-6251 DAQ card, yielding a maximum sampling rate of 200kpsps.

Control of the circuit under test was achieved through an experiment execution engine written in C#, which used the NI DAQmx library of functions that is bundled with ELVIS. By invoking the appropriate API calls, analogue input and output channels of ELVIS were configured and when a request was submitted, a software-generated waveform was injected into the circuit under test and the output captured with an appropriate PCI-6251 analogue input channel.

IV. OpAmp Lab Software Components

a. The OpAmp Lab Server

The Lab Server implements the iLab-B framework's Lab Server API using web services under Microsoft's ASP.NET. The web service is implemented in the VS.NET C# language on a web Server running Internet Information Services (IIS), and can be invoked through a fixed URL reserved for the Web Based Lab. The Lab Server also runs an SQL database under Microsoft SQL Server 2000. This database is accessed from the web services module for authentication and authorization, logging calls to the web service, retrieval of hardware information relating to the experiment, etc. In addition, the database is used to queue experiment requests and to temporarily store any experimental results that were processed at the Lab Server. When an experiment is submitted, the specification is stored in the database, along with a suggested execution priority and a flag indicating that it has not been executed.

An experiment execution engine written in C# ran separately from the Lab Server web service. The experiment engine periodically checks the queue of submitted experiments, and retrieves either the first job in the queue or the one with the highest priority. It then processes the experiment specification provided by the user at the Lab Client, and sets up the hardware appropriately. This step is accomplished by communicating with NI ELVIS DAQ Card and the switching matrix whose probes are attached to the system under test. ELVIS/ DAQ board inputs are set by the experiment engine, which makes calls to a dynamic link library through NI DAQmx library, providing communication with the DAQ board over the USB and PCI interfaces.

b. Client

The Client of the OpAmp Lab was developed using C#. A Windows Form ("Winform") control was developed in C# and embedded in a web page that could only be accessed through the Service Broker after a session has started. The C# Client creates a proxy class for the Service Broker web service pass through methods and this class is used for submitting experiments, retrieving results, and otherwise communicating with the Server or the Service Broker. One objective in developing the interface was to allow the student to carry out activities that are the analogue of connecting nodes in a circuit, and thereby make lab usage more active. This objective was balanced with the need to make the interface reasonably attractive while keeping the control size as small as possible to reduce download times. The size of the final compiled assembly of the Client was 88kB. As shown in Figure 4, the Client presents a canvas with a signal source and an oscilloscope along with impedances arranged around an operational amplifier.

To configure a circuit, the student would successively click on any pair of nodes he wishes to connect at a time. Once all the nodes needed to complete a circuit are connected, the student clicks the "Run" button to submit the configured circuit for execution. On submission, the Client checks the validity of the circuit before forwarding it to the Server. Only circuits that have been validated would be passed on to the Server while others would be returned with an appropriate message to the student.

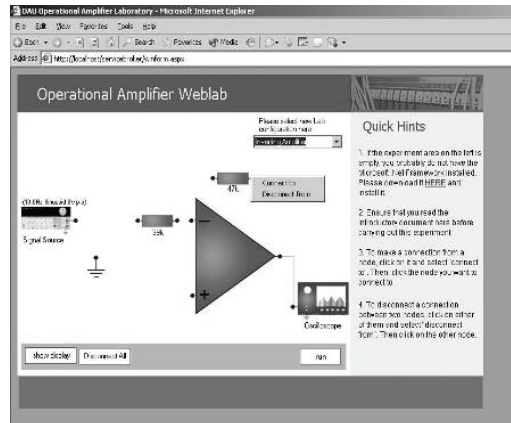


Figure 4: Interface of the OpAmp Lab Client

c. Service Broker

The Service broker tier of OpAmp Lab employs the generic MIT Service Broker distribution 6.1. This allowed the development time to be cut, since web service methods had already been implemented and were simply invoked as needed. This ability to simply install the Service Broker tier is one of the major bonuses of using the iLab architecture. Considering the fact that the bulk of the communication infrastructure of the iLab architecture is implemented on the Service Broker, being able to simply install an existing Service Broker led to large savings in time.

d. Database

In addition to Microsoft's IIS, the OpAmp Lab Lab Server runs SQL Server 2000 for experiment data and configuration management. The SQL Server can be accessed by both the web services module and the experiment engine at any given time. It consists of SQL tables used to encapsulate Service Broker and Lab Server configurations, experiment records, and group management, as well as of a number of procedures used to manipulate them.

For instance, the LS (Lab Server) System Config table stores all Lab Server-specific information, such as the Lab Server's unique ID and the default experiment it should run. In addition, it contains a Boolean flag which is automatically set to true whenever the experiment engine is running, but which is set to false otherwise. This mechanism allows the Lab Server to gracefully reject experiment requests at those times when the experiment engine is not running. Similarly, the Brokers table is used to store data pertaining to Service Brokers registered at the Lab Server. Therefore, it contains information such as the Service Broker's ID, the Service Broker, and Lab Server authentication passkeys, and the URL for the Service Broker-to-Lab Server web service interface.

V. Security

Since ASP.NET is based on HTTP, it becomes possible to leverage the security features available in IIS to provide strong support for standard HTTP authentication schemes. More

specifically, the OpAmp Lab Server makes use of IP filtering to deny access to the Web Lab service from all hosts except the trusted Service Broker. In addition, all connections between Lab Server and Service Broker can be further secured using the Secure Sockets Layer (SSL) protocol. As a result, any data exchanged between these two hosts does not travel as clear text but instead is always encrypted prior to being sent over the internet.

Beyond the lab-specific security considerations outlined above, the iLab framework also provides a simple mechanism based on SOAP headers for authentication. When the Lab Server becomes registered at the Service Broker, the latter assigns it a passkey that is bound to that Lab Server's unique ID. Similarly, when the Lab Server registers the Service Broker, it to o assigns the latter a passkey bound to the Service Broker's unique ID. From this point on, all communication between Lab Server and Service Broker includes a SOAP header parameter containing the passkey for the corresponding host. This mechanism, when used along with SSL to encrypt the payload contents, ensures that only registered hosts are able to successfully execute web service methods on either Lab Server or Service Broker.

VI. Results

OpAmp Lab has been used for two years at Obafemi Awolowo University's Electronic and Electrical Engineering Department. To use the lab, a student would register at the URL of the OpAmp Lab Service broker, <http://OpAmp Lab.oauife.edu.ng> . After registration, students could log on and launch the Client. The Client interface described earlier is presented and the student then connects appropriate nodes to configure the interface as one of six different circuits. For example, Figures 5a and 5b show the interface connected as a non-inverting amplifier and summer respectively. After the student has completed the connection, he pushes the "run" button to submit the request to the Server. On submitting the experiment, the interface automatically changes to the graph screen, on which the input and output waveforms are plotted once the results are returned from the Server.

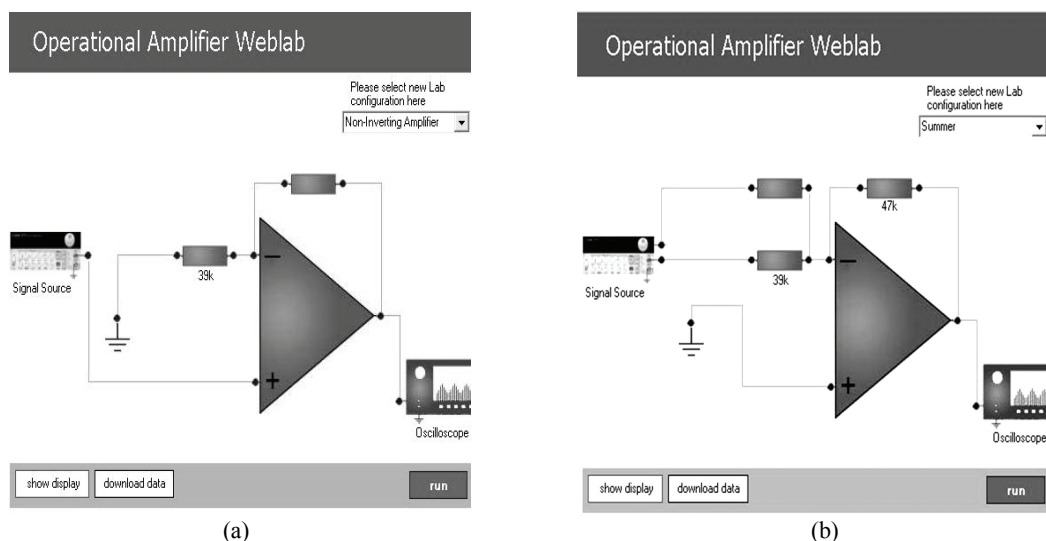


Figure 5: OpAmp Lab Interface configured as (a) a Non-Inverting Amplifier (b) a Summer

it was decided that the wiggles were actually an opportunity to engage students working on the laboratory. Hence, the wiggles were left and one of the questions students were asked after carrying out the experiment was to suggest what caused them.

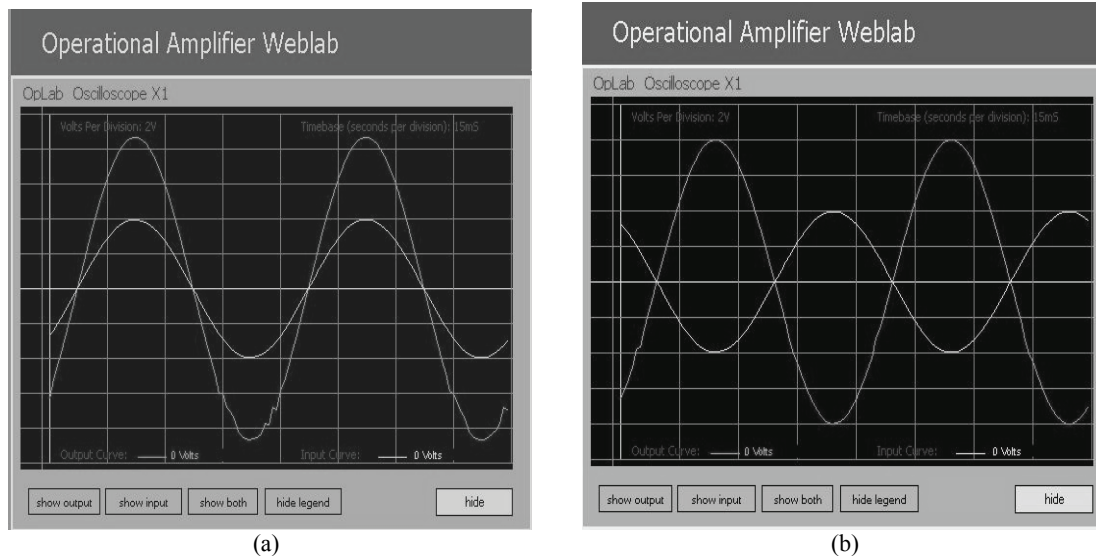


Figure 6: OpAmp Lab experiment results for (a) a Non-Inverting Amplifier (b) a Summer

Evaluating the effect of VLs or any other pedagogical tool has long been recognized as a tricky problem⁸. The question of how effective a tool is to a student inevitably raises issues about the student's ability to correctly assess how much he has gained. A scenario currently favored is to evaluate the real-life performance of students using the new tool over a relatively long period of time. Another technique, often hampered by students' ability to correctly assess themselves (as well as their capacity for honest answers!) is the use of questionnaires. Since OpAmp Lab has only been in use for two years and proper assessments of students' performance have not been carried out, the use of questionnaire-style assessment to determine effectiveness was resorted to. Each student who used the lab was presented with the following list of ten statements after completing his experiment:

1. The virtual lab will be useful for students
2. There are no ways in which virtual labs are better than real
3. Virtual labs will eventually replace real labs
4. Using the labs was very enjoyable
5. The lab is very intuitive to use
6. Using the lab was not intellectually stimulating
7. The lab obviously needs major improvements
8. The response of the lab was obviously slow
9. The interface/arrangement of elements is confusing
10. Using the lab has helped me in at least one way

The statements were deliberately crafted so that there were five statements with a negative sense (statements 2, 6, 7, 8 and 9) and five with positive sense (statements 1,3,4,5, 1nd 10). Students were asked to rate each statement with a number as follows:

- 1=I strongly agree
- 2=I agree
- 3=I do not know/I am not sure
- 4=I disagree
- 5=I strongly disagree

Two groups of students have taken this exercise (group sizes 23 and 58). The averages of their scores for each question as well as the standard deviation per question are presented as table 1 and Figure 7.

Table 1: results of students' ranking of assessment statements

Statements	1	2	3	4	5	6	7	8	9	10
S.D	0.3	1.3	1.2	0.5	0.5	1.4	1.0	1.1	0.5	0.7
Average	1.1	2.9	3.0	1.3	1.4	3.3	2.2	3.6	4.3	1.7

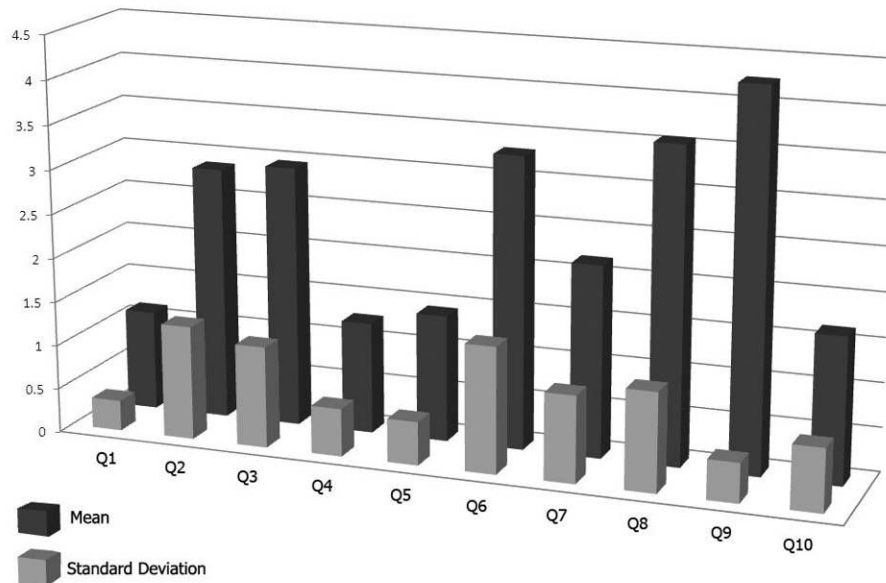


Table 7: results of students' ranking of assessment statements

VII. Discussion

The data indicates that students' perception of the lab was generally positive, even though they feel there is still room for improvements (statement 7). This was a very valuable result, indicating as it does, that VLs would be very well received by students as a viable supplement to "real" labs in Nigeria. It should also be noted that the only statement where OpAmp Lab got a relatively high negative (statement 6) also features large deviations in the student's scores, suggesting that personal inclinations played a strong role in students' response here. It was interesting to note that the students strongly felt the interface was effective (Statement 9). Their response to this statement was both one of the most positive to any statement, and one of the most consistent (low deviation). It was believed that their good

impressions about the interface were largely responsible for the student's positive impressions of the entire lab and a simple scheme was developed whereby this hypothesis could be confirmed from the data in hand.

An Approximate Lab Effectiveness Index of the lab was developed by averaging the student's scores for statements 1, 4 and 10. This index was interpreted as being indicative of the level to which the student believed the lab had been useful to him, and would be useful to others. Likewise, by averaging scores for statements 5 and 9, a User Interface Effectiveness Index was computed for each student, meaning that from these two statements, an idea of how effective the student felt the lab user interface was could be extracted. In developing the User Interface Effectiveness Index, it was important to align the sense of the two questions (statement 5 was made in a positive sense while statement 9 was in a negative sense). To do this, the scores for statement 9 were inverted such that the statement effectively became "*The interface/arrangement of elements is **not** confusing*". Therefore, a score of 4 (disagree) for the previous statement 9 for example, was now converted to a 2 (agree) for the re-aligned statement 9.

The Lab Effectiveness Index was correlated in turn with three parameters which could have affected the student's perspective. The parameters chosen were student's perception of: the responsiveness of the lab (obtained from statement 8), the intellectual challenge posed (from question 6), and the effectiveness of the interface (obtained from the User Interface Effectiveness Index). The sense of statement 6 was re-aligned to a positive sense as described above for statement 9. The result generated is shown in table 2.

TABLE 2: Correlation of students' satisfaction with OpAmp Lab with:

Responsiveness	Interface	Challenge
-0.13	0.41	0.24

Even though it was expected that interface design would contribute to students' positive or negative perception of OpAmp Lab, it was still interesting to see that student's perception correlated quite well with how effective they thought the user interface was. More work needs to be done in this area to determine what aspects of the interfaces were well received. Also important was the fact that students are more likely to feel the lab is effective if it offers an intellectual challenge than if it is very fast. This flies in the face of the general belief that students would not want to work with a lab unless it is very responsive.

VIII. Conclusion

OpAmp Lab, a pedagogic operational amplifier virtual based on the MIT iLab-B architecture has been developed and used at Obafemi Awolowo University for two years. The circuit under test is build around the Dozen Impedance operational amplifier configuration and deployed on an NI ELVIS. Usage data so far, generated from questionnaires reveals that students are very happy with the lab, and indicate that the lab and similar labs could be a good way to improve experimental components of the education of Nigerian students. Data also reveals that the interface is more important than how responsive the lab is, or how

intellectually challenging it is. More conclusive data would be generated in future from actual students' performances after using the lab.

IX. Acknowledgement

The authors would like to thank Jesus del Alamo, Steve Lehman, Jud Harward and members of the Center for Educational Computing Initiatives (CECI) at Massachusetts Institute of Technology for assisting with various aspects of the research. Some of the components used for the research were donated by Maxim Semiconductors. The research was funded by a grant from the Carnegie Corporation of New York.

Bibliography

1. Del Alamo, J. A., L. Brooks, C. Mclean, J. Hardison, G. Mishuris, V. Chang, And L. Hui, "The Mit Microelectronics Weblab: A Web-Enabled Remote Laboratory For Microelectronics Device Characterization." 2002 World Congress On Networked Learning In A Global Environment, Berlin (Germany), May 2002.
2. Del Alamo, J.A., Chang, V., Hardison, J., Zych, D., and Hui, L.(2003): An Online Microelectronics Device Characterization Laboratory with a Circuit-like User Interface, Proceedings of the International Conference on Engineering Education, Valencia, Spain.
3. Del Alamo, J.A., Hardison, J., Mishuris, G., Brooks, L., McLean, C., Chan, V., and Hui, L. (2002): Educational Experiments with An Online Microelectronics Characterization Laboratory, Proceedings of the International Conference on Engineering Education, Manchester.
4. Viedma, G., Dancy, I.J., and Lundberg, K.H.(2005): A Web-Based Linear-Systems ILab , Proceedings Of The American Control Conference,. Vol. 7, Issue 8-10, pp 5139 – 5144.
5. Hardison, J., Zych, D., Del Alamo, J.A., Harward, V.J., Lerman, S.R., Wang, S.M., Yehia, K., and Varadharajan, C. (2005): The Microelectronics Weblab 6.0 – An Implementation Using Web Services And The ILab Shared Architecture, *Proceedings Of The Conference On Exploring Innovation In Education And Research*, Tainan, Taiwan.
6. Kehinde, L.O. (1989): The “Dozen-Impedance” Operational Amplifier Module for Experimentation. International Journal of Electrical Engineering Education, vol. 26, No. 3, 1989, pp 224-232, Manchester UP.
7. NI ELVIS User Manual, National Instruments, August, 2002.
8. Bowman, D., Gabbard, J., and Hix, D. (2002): A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods. *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 4, 2002, pp. 404-424