

Implementation of Time and Frequency Response Analysis for Web-Based Laboratories

Teyana SAPULA
Telecommunications Eng., University of Dar Es Salaam
Dar Es Salaam, Tanzania

and

Damian HAULE
P.O. Box 35131
Dar Es Salaam, Tanzania

ABSTRACT

The University of Dar Es Salaam has developed the web-based laboratory for Time and Frequency Response Analysis. The purpose of this web-based laboratory is the utilization of real data from real experiments, in terms of instrumentation and experimental circuits, rather than simulations. The use of web-based laboratory came after realizing the difficulties imposed by the traditional laboratories. Web-based laboratories allow students and educators to interact with real laboratory equipment located anywhere in the world at anytime. This paper presents the implementation of web-based laboratory of single stage common emitter, resistor capacitor coupled amplifier using National Instruments Educational Laboratory Virtual Instrument Suite platform. Two components are deployed: time response analysis and frequency response analysis. The experiment allows students to carry out time and frequency analysis of the amplifier. The modular can be used to any microelectronic circuits to carry out any time response and frequency response analysis. Both the time response and frequency response analysis results of the amplifier are validated.

Keywords: Batched iLab Shared Architecture, Client, Lab Server, NI ELVIS, Service Broker.

1. INTRODUCTION

Web-based laboratory means online experimentation on real processes. Contrary to simulations, which rely on mathematical models, remote laboratories deal with real signals. Laboratory experiments provide students with practical experience that help them better understanding the theory taught in classes. However, traditional laboratory instruments are usually expensive such that many educational institutions cannot afford the instruments they require for their students. Sometimes students are overcrowded in laboratory sessions. In addition, laboratory personnel need to be hired to operate the facilities, thus imposing additional costs.

The goal of the web-based is to remove the problems imposed by the traditional systems and give hands-on experiences in real hardware, in real time. Web-based laboratories enable the students and educators to access experiments any time from anywhere through the internet. By providing remote access to laboratory to students, the problem of costly traditional laboratories can be overcome by using the few laboratory resources available at that institution. In addition, web-based labs will increase the range of experiments available at institution as not only the students will use the experiments at their institutions but they will be able to share the experiments with other educational institutions. Moreover, web-based laboratories will be suitable to open universities and other distance learners to enable them to get the hands-on, real-time experiences. Furthermore, students will experiment with freedom at their own time and have relevant experience. Developments of such laboratories are useful in developing countries where funds for education resources are hardly available.

The web-based laboratories existing over the decade now, but many of educational institutions use expensive equipment which is difficult for the developing country to deploy. In addition, many of available online laboratories are ad hoc systems and tailored for certain laboratory devices [1].

This paper describes the implementation of web-based laboratory of a single stage CE, RC coupled amplifier. The objective of this experiment is to enable students to carry out time domain and frequency domain analysis of different microelectronics circuits online. The National Instruments (NI) provided the affordable Educational Laboratory Virtual Instrumentation Suite (ELVIS) kit which interfaces the different components with the web server. Graphical User Interface (GUI) developed using Java facilitates the on-line access and control of experiment parameters. The architecture used is known as iLab (internet laboratories) developed by Massachusetts Institute of Technology (MIT) iLab team in 2002. The iLab offers unifying software framework, which supports single sign-on online access to a wide variety of laboratories.

2. SYSTEM ARCHITECTURE

The architecture used is client-server architecture consists of Lab Server, Service Broker and client [2], as shown in Figure 1. First tier, **Lab Server** has the connection to the real laboratory device. It executes experiments requested by Service Broker and notifies Service Broker when results are ready [3]. It knows nothing about the students using the system, but it stores temporary experiment specifications and results. MIT started iLab development from batched experiments, where user defines all parameters at a time for lab device before running. MIT has released an interactive version of the iLab, where user can specify parameters from a fly to the Lab device, but comparing to the batched version it requires more effort to develop and wider bandwidth for efficient use. Therefore, batched experiments are more practical alternatives in Tanzania and other countries with low bandwidths and unstable electricity. Additional benefits for batched experiments are that those are not that sensitive for power cuts and do not require Service Broker or Lab Server time during specifying parameters for the experiment running.

Service Broker as middle tier is responsible for authentication, authorization and to forward the communication between Client and Lab Server as well stores students experiment specification and results under its account. It offers user interface for administration task (for example setting permission or adding new Lab servers to the system) and it is the first place where student comes to require permission to a lab.

Third tier, **Client**, provides user interface for student to specify parameters for running experiments. Architecture supports Java application, applets or html-based clients. Client is downloadable from Service Broker and running on students own machine. Especially with batched experiment, student can set up values without using Service Broker or Lab Server time.

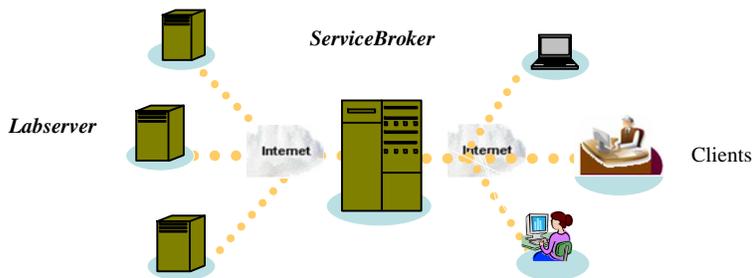


Figure 1: Overview of the batched experiments architecture

3. IMPLEMENTATION OF RC COUPLED AMPLIFIER

The single stage CE, RC coupled amplifier experiment is constructed in NI ELVIS prototyping board. The Data Acquisition Card (DAQ) is PC 6251. First, the input and output voltages are determined (time response analysis), followed by frequency response analysis.

Time Response Analysis (Transient Analysis)

Time response analysis of the single stage CE, RC coupled amplifier is a measure of the input and output voltages. The transient analysis uses Function Generator (FGEN) and Oscilloscope (SCOPE) instruments. The input to the circuit is connected to the FGEN and the SCOPE is connected to the output. Figure 2 is the experimental setup of single stage CE, RC coupled amplifier. To create the experiment setup, the image of the circuit is loaded on the client's window. Once the image is loaded, the Transient Analyze is chosen. Then, the students set the experimental input specifications. The input specifications are then sent to the Lab Server to perform the actual experiment through the Service Broker. When the experiment is done on the hardware, the results are passed back through the Dynamic Link Library (DLL) to the setup and then call to the experiment engine. The experiment engine stores the results in the Lab Server database in the ExperimentResultXML file. The experiment engine then notifies the service broker that the results are ready. The service broker fetches the results from the Lab Server database and passes to the client where the results are displayed to the student. In the Lab Server, the processing of experiment is divided into two main parts: LabView and Visual Basic.

LabView

Time response analysis in LabView has three main parts:

TransientAnalyze.vi: This is the main entry to the LabView code. This is the first class called by the compiled DLL from the Visual Basic in the Lab Server visual basic. This module passes clients' parameters to the FGEN.vi

FGEN.vi: This VI calls the various hardware instruments. It calls function generator and Data Acquisition (DAQ) card.

RunFGen.vi: Calls function generator and DAQ functions. The function generator generates the requested waveform and the DAQ to sample the analogue signals.

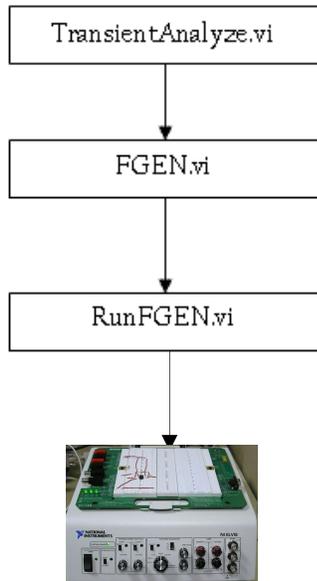


Figure 2: The Time Response Analysis hierarchy

Visual Basic

Visual basic contains of eight components. But the main components are:

Experiment_engine: Checks the queue, if any experiment, de-queues it. Then it calls the setup, which has parseXMLSpec method and parses the experiment specifications. Then the setup calls the terminal that creates the terminal from the specific terminal class to get the specific parameters of that instrument. When the experiment is done on the ELVIS hardware, results are returned back through the LabView, to the experiment_engine. The data points are then put into an XML file called the “Experiment Results.” and sent back to the client for display to the client.

Setup: This class has the method parseXMLSpec. This method parses the experiment specification (an XML file that contains the experiment parameters chosen by the client. It parses the XML parts from validation engine and delegates each parsing of each terminal to Terminal class. The parsed data elements are loaded into class variables for processing by other private and internal methods. The setup will determine whether the experiment being run is time response or frequency Response. This is done depending on the parameters send by the client and the instrument selected by the client whether it is a FGEN or BODE.

Wrapper: This class provides a wrapper around the LabView dlls that communicate with the ELVIS board for the ACAnalyze experiment. It runs an experiment that is in the setup (assumes that setup is validated before passing to this function). It returns an arrayList with the waveform values generated by running the experiment. The RunExperiment() method in the TransientAnalyze class calls the compiled LabView DLL with the specified parameters. The DLL runs the experiment on the ELVIS hardware. Once the experiment is run, it returns from the TransientAnalyze class back to the runExperiment() method in the experiment engine with an array of data for graphs that will be displayed to the client. The data

points are then put into an XML file called the “Experiment Results” and sent back to the client for display to the client.

Validation_engine: This is the first thing that is called before the job is queued for execution. It checks whether the inputs specified by the user meets the specification set by the designer of the experiment when setting up the assignment. It works the same way as the parseXMLSpec() method in the execution engine to extract the experiment parameters and checks these values against the values stored in the database.

Lab Server Administrative Interface and Database:The Lab Server administrative interface is an active server page (ASP) website where experiments are created. It interacts direct with the structured quel language (SQL) database.

Client as a Third Tier

The client is where the students/users specify the parameters to be used in the experiment. It is a Java Applet launched from the service broker. It uses Simple Object Access Protocol (SOAP) to communicate with the service Broker as shown in Figure 3.

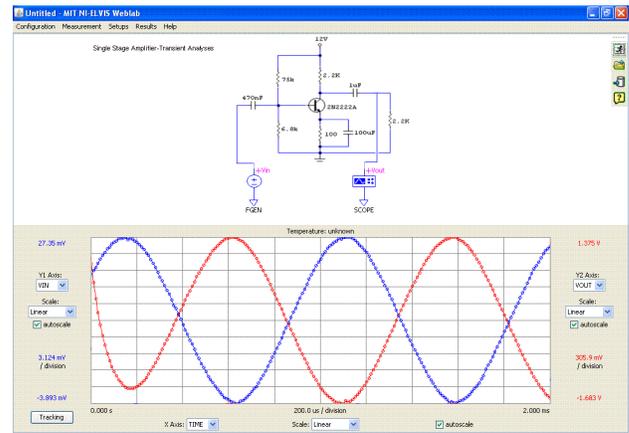


Figure 3: Input and output voltages of the single stage amplifier and its waveform. The input is shown in blue and output shown in red

An Example: Frequency Domain Analysis (AC Analyze)

The Bode Analyzer is used to display frequency response and the corresponding phase angle (Bode Plot) of the circuit. The magnitude against frequency and phase angle against frequency are obtained by making use of sweep feature of function generator and analogue input capability of DAQ device. The same circuit used in transient analysis is used in frequency analysis. Figure 4, shows the waveforms obtained when testing the frequency response in NI ELVIS.

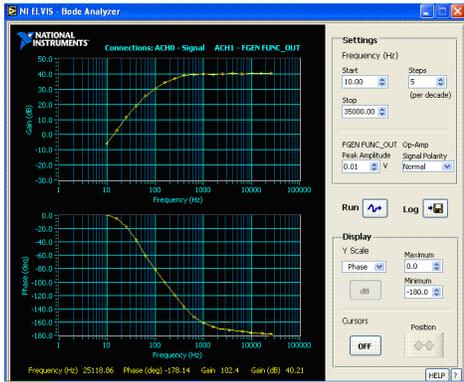


Figure 4: Gain and phase of single stage amplifier in NI ELVIS

After that, the virtual instrument.vi is created in LabView. The hierarchy of how the frequency response of a single stage amplifier is created in LabView is illustrated in figure 5 and figure 6. LabView in frequency response has two stages: BodeAnalyzer.vi and ACAnalyzer.vi.

BodeAnalyzer.vi is first created in the LabView. This vi runs Bode Analyzer parameters from the client and the function generator hardware to sweep the sine waves as shown in figure 5. The BodeAnalyzer.vi utilizes the Bode Analyzer VI Express which is provided in LabView to run the ELVIS functionalities. This vi is then put in ACAnalyzer.vi as a subvi.

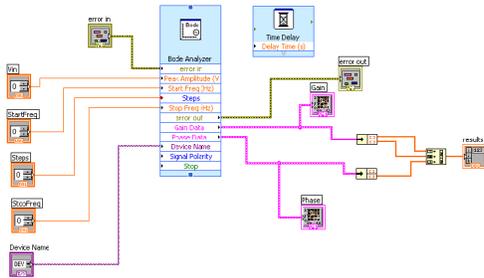


Figure 5: The BodeAnalyzer.vi

The ACAnalyze.vi is the entry point to the LabView from the DLL. This enables the parameters from the client to get to BodeAnalyzer.vi, which runs the experiment on the ELVIS board as shown in figure 6. When the experiment is done, the results are collected in the result waveform in the BodeAnalyzer.vi and passes back to ACAnalyze.vi to DLL.

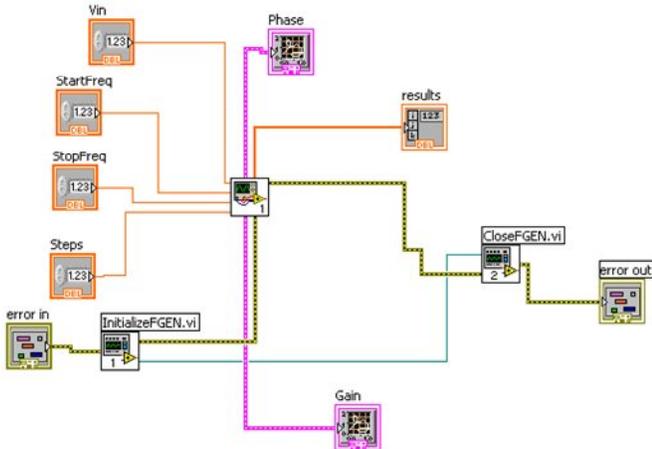


Figure 6: TheACAnalyze.vi

In visual basic, the same path used in transient analysis, is the same in frequency analysis. As in transient analysis, the client launches the single stage amplifier with the bode analyzer parameters. These parameters are amplitude, start frequency, stop frequency and steps. The results are the gain in dB against frequency in logarithm scale and phase in degrees against frequency in logarithm as shown in figures 7(a) and 7(b) respectively.

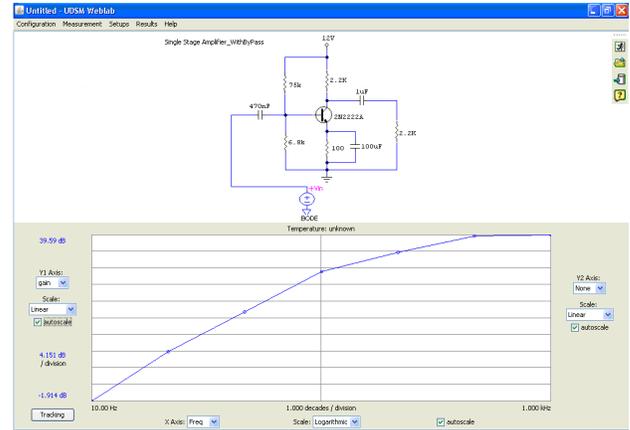


Figure 7(a): The gain against frequency

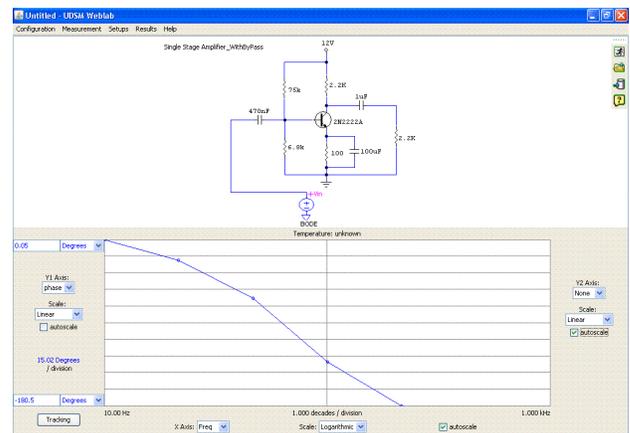


Figure 7(b): The phase against frequency

There is a debate on the technical value produced by the web-accessible labs. Figure 8 shows the comparison of traditional, simulation and web-based laboratories.

Results of the three types of laboratories: traditional, simulations and web-accessible of the amplifier without the bypass capacitor.

Freq(Hz)	Vin (V)	Traditional		Simulations		Web-Based	
		Vout (V)	Gain(dB)	Vout	Gain	Vout	Gain
10	0.007	0.003	-5.3	0.03	-7.0	0.041	-4.6
100	0.007	0.007	26.4	0.16	27.0	0.16	27.0
1k	0.007	0.204	29.3	0.45	28.1	0.229	30.3
10k	0.007	0.617	38.9	0.48	40.1	0.684	39.8
35k	0.007	0.668	39.6	0.64	40.7	0.716	40.2

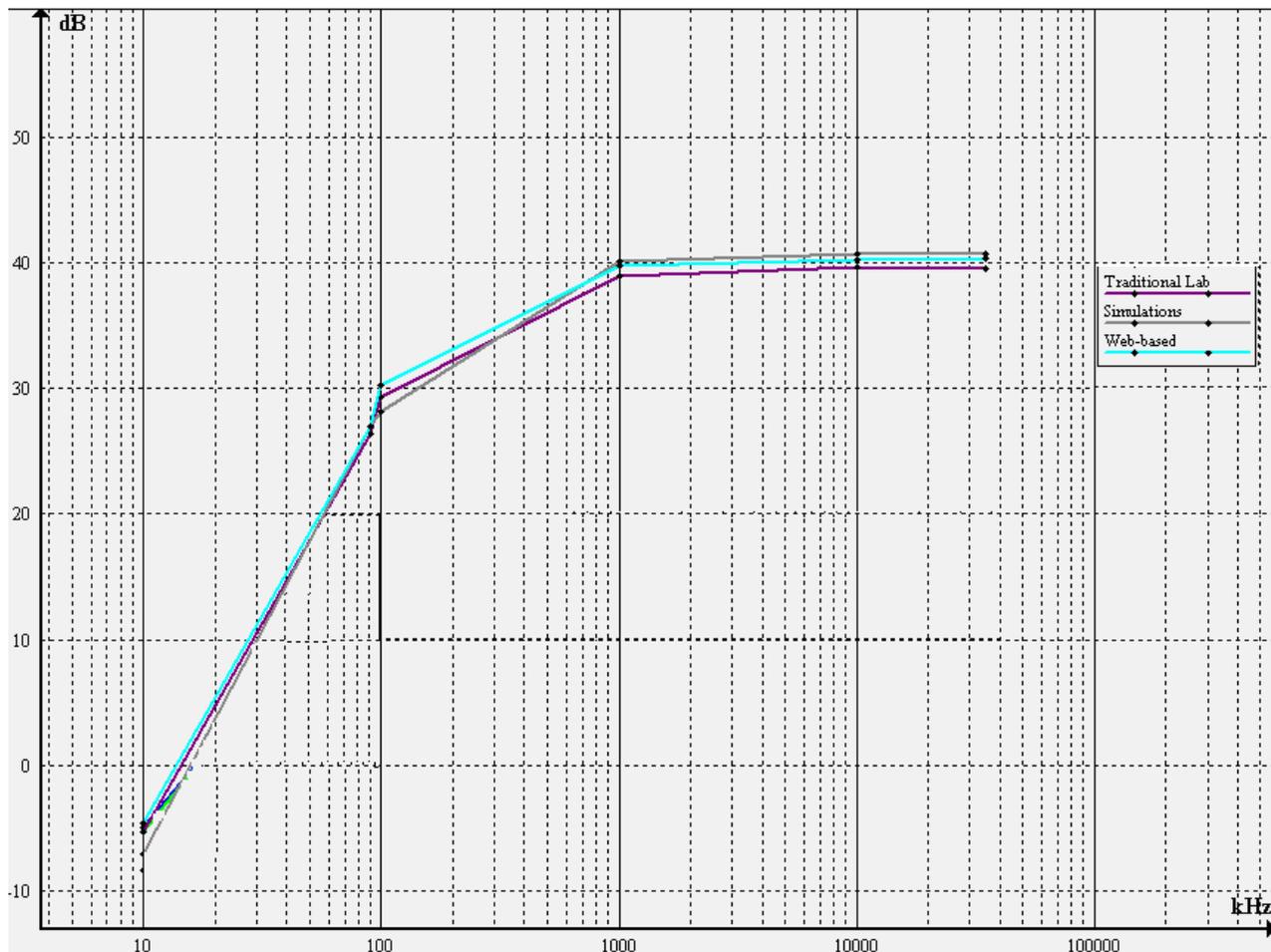


Figure 8: Comparisons of traditional, web accessible and simulations results

4. CONCLUSION

As many educational institutions have difficulties in providing hands-on experiments to their students due to many limitations imposed by traditional laboratories, the presented web-based laboratory is intended to provide students with the same interaction to real hardware as traditional ones. The presented paper is an effort to explore the capabilities of web-based laboratories in science and engineering education. The aim is to make students to get the same quality of education using user friendly environment. The effort is underway to make sure that many instruments are deployed in the modular so that students can explore many real experiments without hindrance of laboratory equipment. In addition, with web-based laboratory, e-learning institutions will be able to provide hands-on experiments to their students.

REFERENCES

- [1] C.C. Ko, B. M Chen, J. Chen, Y. Zhuang and K. C. Tan, "Development of a Web-Accessible laboratory for control experiments on a coupled tank apparatus," **IEEE Transactions on Education**, vol. 44, #1, pp. 76–86, 2001.
- [2] J.V. Harward, J. del Alamo, V.S. Choudhary, K. deLong, J. Hardison, S.R. Lerman, J. Northridge, D. Talavera, C. Wang, K. Yehia, D. Zych, " iLab: A Scalable Architecture for Sharing Online Experiments", **International Conference on Engineering Education**, Gainesville, Florida, 2004.
- [3] P. Bailey, "The online experiments shared architecture and the future of web based laboratory Experiments", 2004.
- [4] V.N. Gerardo, "Design and Implementation of a Feedback Systems Web Laboratory Prototype", **AUP Final Report**, MIT EECS, 2004.
- [5] G. Viedma, J. D. Isaac and H. L., Kent, "A Web-Based Linear-Systems iLab" **submitted American control conference**, 2005.

- [6] S. Gikandi, "A Flexible Platform for Online Laboratory Experiments in Electrical Engineering", **Masters of Electrical Engineering and Computer Science**, Massachusetts Institute of Technology, 2006.

BIBLIOGRAPHY

- [1] S.M., Bianchi, S.M., Berno, , , B.G., Marcelo, D.S. Jr Rosevaldo, A.G., Luis, "Collaborative virtual lab for electronic experience", **Proc. International Conference on Web Based Education**, Innsbruck, Austria, 2004, pp.86-88.
- [2] C.L., Hsiung "An internet-based graphical programming tool for teaching power system harmonic measurement", **IEEE Transactions on Education**, vol. 49, Issue 3, 2006, pp. 404-414.
- [3] A. Khamis, M. Peter Vernet, K. Schilling, "A remote experiment on motor control of mobile robots", **Proc.of the 10th International conference on Control and Automation**, Lisbon, Portugal, July 2002.
- [4] A. Ferrero, S. Salicone, C. Bonoranand M. Parmigiani, "ReMLab: a Java-based remote, didactic measurement laboratory", **International Symposium on Virtual and Intelligent Measurement Systems**, 2002.
- [5] M. Exel, S. Gentil and D. Rey, "Simulation workshop and remote laboratory: two Web-Accessible training approaches for control," **in Proceedings of the American Control Conference**, Chicago, 2000.
- [6] J. Hardison, "Increasing reliability, reusability and measurement flexibility in the Microelectronics weblab", 2006.
- [7] M. Gilbert, J. Picazo, M. Auer, A. Pester, J. Cusido. J.A. Ortega, "80C537 Microcontroller remote lab for e-learning teaching," **International Journal of online Engineering**, vol.2, 2006.