DEVELOPMENT OF *VirCEL*: A VIRTUAL COMMUNICATION ENGINEERING LABORATORY

Aderemi A. Atayero, Ayodeji V. Ogundeji, Michael O. Adedokun

Covenant University (NIGERIA)

Abstract

Access to quality hands-on experience is an essential part of engineering instruction. The laboratory experience is a very vital part of engineering education. Availability of this required access is however not always given. Through laboratory hands-on experiments, students are able to develop practical skills such as real-world problem solving, design, communication, teamwork, independent thinking and creativity, while working in the laboratories. These laboratories with physical presence of students at the experimental rigs are referred to as the hands-on or traditional laboratories in most of the literature. But with the advances in ICT during the last three decades, the simulated and virtual environments have opened up some very innovative techniques in teaching. Such environments have been explored to meet the objectives of laboratory work. This has given rise to the development of laboratories referred to as virtual laboratories. This class of laboratories is essential and come with various advantages, the least of which is not the ability of learners to study at their own convenience in time and space. This paper presents a detailed step-wise methodology for the development of one of such virtual laboratories for communication engineering on the *LabVIEW*™ platform.

Keywords: VLE, Virtual Laboratory, LabVIEW, Flipped classroom, GUI, BSD, DLL.

1 INTRODUCTION

A virtual engineering 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. Also virtual laboratory makes it possible for students to do experiments selectively, which gives them a new way to put more energy into 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 virtual laboratory is designed to introduce students to experimentation, data-gathering and scientific interpretation early on in their careers.

1.1 Previous Works

In the past decade or so, experimental simulations have appeared with discipline-specific texts, such as [1, 2], or on the World Wide Web at sites such as the Physical Sciences Resource Center [3]. Many of these simulations are implemented using Java applets, and are well designed and executed. Other online experimentation include the Stevens Institute of Technology in their Remote Dynamical Systems Laboratory [4] which incorporated several online experiments, but is subject to limited use and limited data. Mercer University has implemented the Online Interactive Chaotic Pendulum [5]. The site has an exemplary interface and data presentation, but it does not provide easy access to data. Several mechanical engineering courses have online experimentation, including Curtin University [6].

These sites tend to be highly technical and their online availability appears to be limited. The Indian Institute of Technology Kharagpur has successfully implemented an online Digital Signal Processing laboratory; their aim is to provide a virtual laboratory platform for engineering undergraduate students studying the course of Digital Signal Processing [7]. This site implements an interactive virtual laboratory architecture with several important signal processing concepts, but access to the virtual experiments is limited to students of the university only. In this work, we present the process of developing online experiments, using a step-wise approach with access solely via the Internet, for a typical engineering laboratory, and specifically a communication engineering laboratory. It presents a discussion on the design and implementation of a virtual laboratory through the use of web-based deployment. The goal of this project is to develop reliable, understandable, repeatable and low maintenance virtual experiments. This work also describes choice of the software selected, benefits of

the software tool chosen for the present case, and steps in the virtualization of one experiment as an illustrative example.

1.2 Choice of Development Platform

The current perception among the engineering disciplines is that multimedia tools are appropriate only for the courses, which offer very little individualized help for students, and therefore, not suitable for the engineering subjects, which require technology-intensive hands-on experience. However, recent developments and applications, specifically the LabVIEW-based applications, have shown that many lecture-based engineering courses and conventional experiments (which are heavily dependent upon specialized instruments) can be updated and integrated with custom-written virtual instrumentation (VI), and can be delivered by computers.

In addition to this, the courses and experiments can be delivered remotely without having multiple copies of the experimental setups. Additional scientific visualizations and advanced analysis can also be added in the form of virtual instruments with minimal cost, which is limited or not possible in the conventional laboratory practice [8]. Moreover, the virtual instrumentation approach is open to further improvements and developments, which may increase the student participation and enthusiasm while providing and ideal delivery environment.

Because of its overall versatility as an engineering tool, the software package LabVIEW is chosen in most of the engineering problems. It is a graphical programming language that allows engineers and scientists to develop their own virtual instrument, which is flexible, modular and economical. Furthermore, the software meets most of the software selection criteria, and it not only does the data manipulation, analysis, and control, but also has some multimedia authoring capabilities with the help of the add-on tools [8]. The LabVIEW software allows for the creating of application-specific templates (sub-virtual instruments) to reduce the production time for the identical subjects. Many useful functions can be incorporated with the LabVIEW programs to perform very useful tasks in a laboratory virtual instrumentation system design. Some of these useful functions [8] are:

a) Adding warning/message sounds or voices;

b) providing instructions, pre-practical tutorial and/or interactive short-tests [9];

c) Generating a test report or a data file in a common text format and

d) Providing GUI that mimics the real instruments.

2 SYSTEM ARCHITECTURE

The virtual laboratory environment mainly focuses on the theoretical demonstrations of experiments. Students are able to access it through a web browser and operate downloaded GUI's embedded in web pages to conduct virtual experiments. Simulation application programs on the server processes user commands and returns the simulated results to them to complete the virtual experiment. The virtual laboratory is based on a *Browser–Server–Database* (BSD) architecture, which consists of three modules: client, web server, database as shown in Fig.1. The web server is used to provide World Wide Web service and to generate the virtual experiment platform embedded in a web page, the database is used to store user information and experimental data [10].
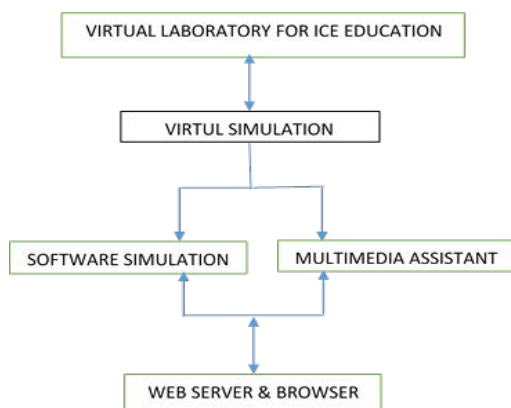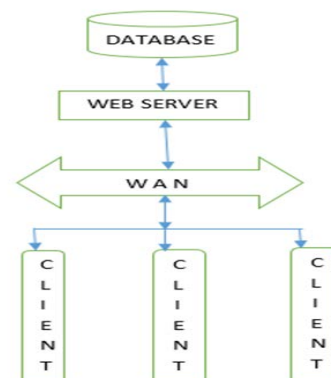

Fig.1. System Architecture.

Fig.2. Hardware Architecture.

3 STEPWISE METHODOLOGY FOR CREATING VIRTUAL INSTRUMENT (VI)

A. Open the LabVIEW application, from the LabVIEW splash screen, choose New VI. This will open up two windows. The gray window (Fig.3) is referred to as the front panel and will be the user interface, and the white window (Fig.4) is referred to as the block diagram, where the graphical source code will be created.
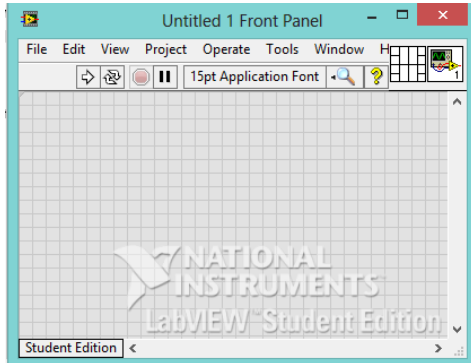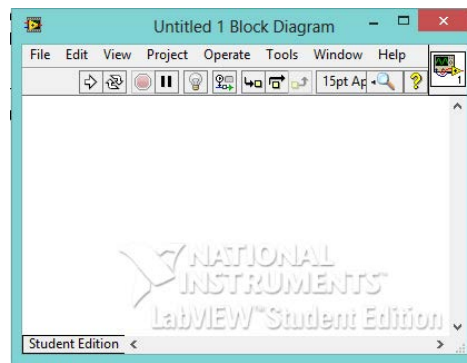

Fig 3. Front panel interface of a VI

Fig. 4. Block diagram interface of a VI.

B. Click on the front panel window to design the graphical user interface, by selecting several inbuilt LabVIEW objects i.e. controls and indicators (Fig.5) from the controls palette.
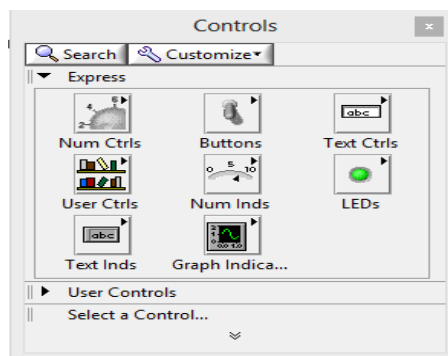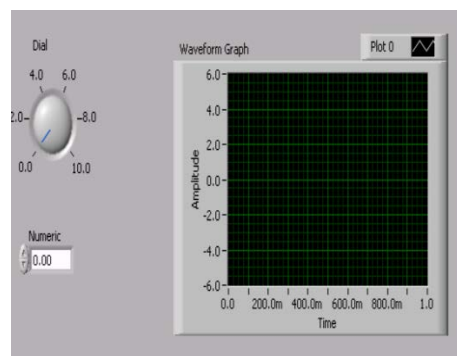

Fig.5. Controls and indicators palette

Fig.6 GUI on a waveform generator [11]

C. The next step is to create the initial block diagram code, by selecting various LabVIEW functions that can be combined to create any custom application. The functions palette has replaced the control palette in the front panel because it consists of all the LabVIEW functions which are required to build any application.
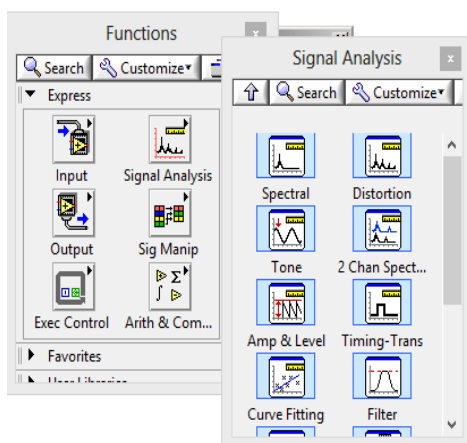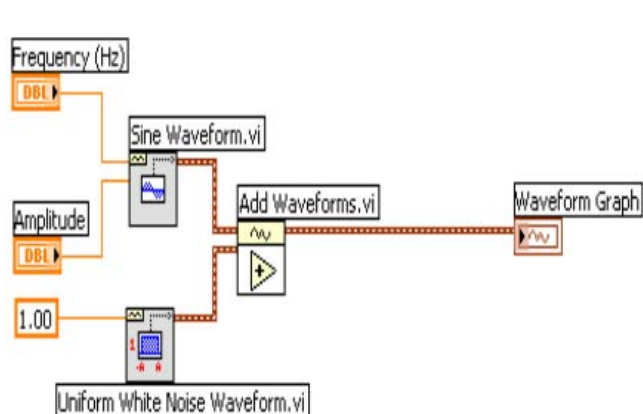

Fig.7. Functions palette.

Fig.8. Sample graphical code of a waveform generator

D. Link the selected objects and functions necessary to create the VI. At this point we have created a working VI. If everything is wired properly, you will notice a white run arrow button as the first button in the toolbar at the top of the front panel and block diagram windows. If the run arrow is solid white, it means there are no errors in the code, but if it appears broken, it means there are errors and there is need for debugging the code.
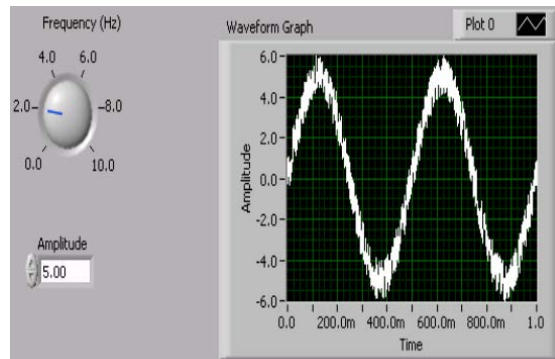

Fig. 9. Example VI of a waveform generator [11]

4   SYSTEM DEVELOPMENT AND IMPLEMENTATION

The VLE is executed in 3 steps [12, 13]: pre experiment, experiment and post experiment. The pre-experiment stage is mainly for obtaining the necessary background information on the project and downloading GUI's for the experiment platform to display the simulation of the experiments. The experiment is executed by adjusting the knobs and imputing value into the simulated experiment and viewing the corresponding changes in the graphical interface displayed. The post experiment step finishes with a Q&A test and an experiment report about the project. Figure 3 shows the experiments working procedure. Before each experiment is conducted, the student user goes through the experiment theory and procedure. The student is able to observe changes in the virtual plot when varying the control values and pushing buttons and adjusting knobs on the instrument panel.

This architecture is based on the built in web publishing tool [14] in LabVIEW, which makes use of the LabVIEW web server to create HTML documents and install front panel images and VI applications in a webpage. The development procedure is as follows. First, a standalone application program is developed with LabVIEW LabVIEW programming. After running a test, it is embedded in a webpage 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 server, implementing the communication between the user and the server, sends the experimental results in a webpage 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 interface Toolkit [15] 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 model parameters and view the output data of the Simulink model in LABVIEW. In conclusion, the webserver and browser architecture suits the thin client system, the client just sends experimental commands to and receives experimental results from the webserver instead 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 browser has to have powerful calculation capability, wide transmission bandwidth and high multithreading safety.
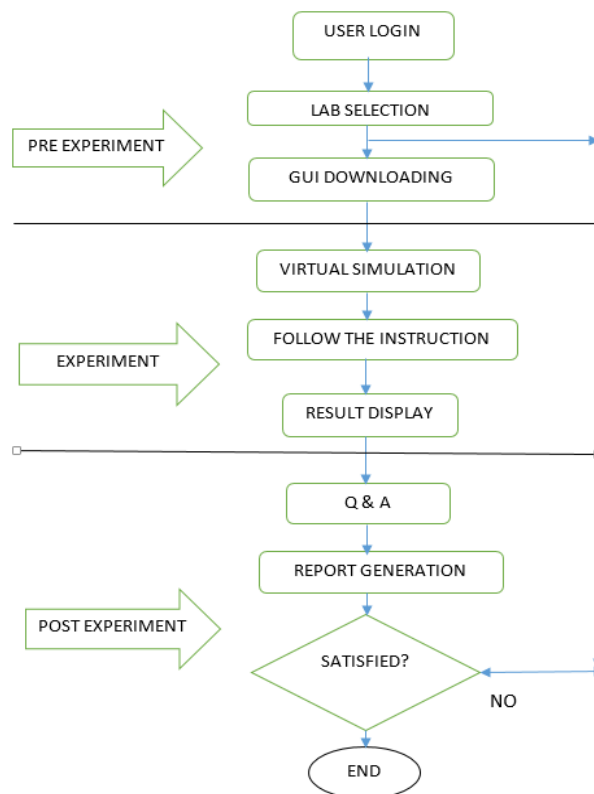
Fig.10 Working procedure of VLE.

5   SIMULATION OF SAMPLING AND SIGNAL RECONSTRUCTION

The objectives of this section are as given below:

- To demonstrate the time domain sampling of band-limited signals (Nyquist theorem).
- To demonstrate the time domain sampling of non-band-limited signals and antialiasing filter.
- To demonstrate the signal reconstruction using zero-order hold and first-order hold filters.
- To demonstrate the sampling in frequency domain (Discrete Fourier Transform).
- To demonstrate the spectral analysis using Discrete Fourier Transform.
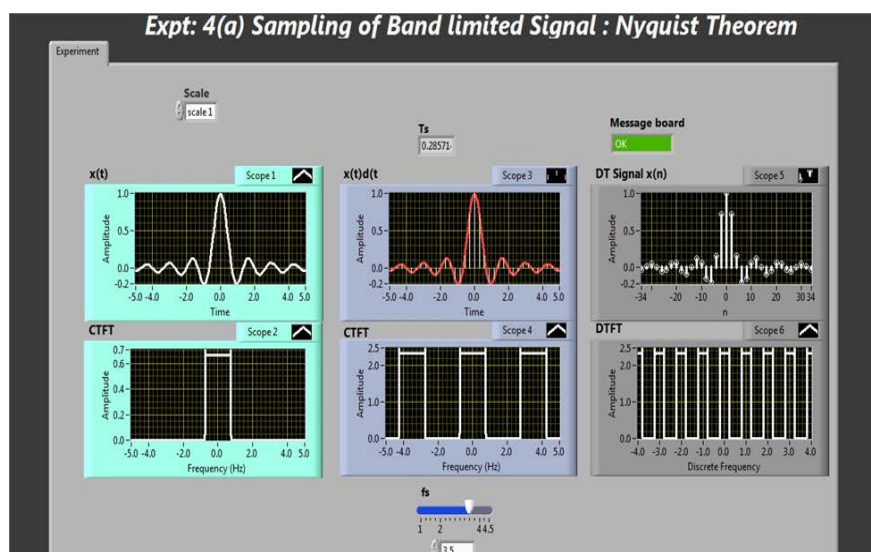

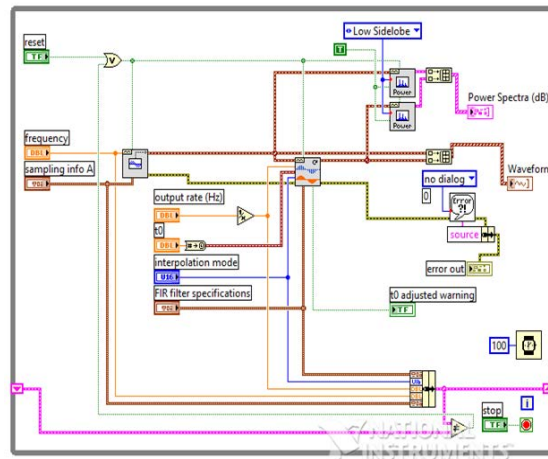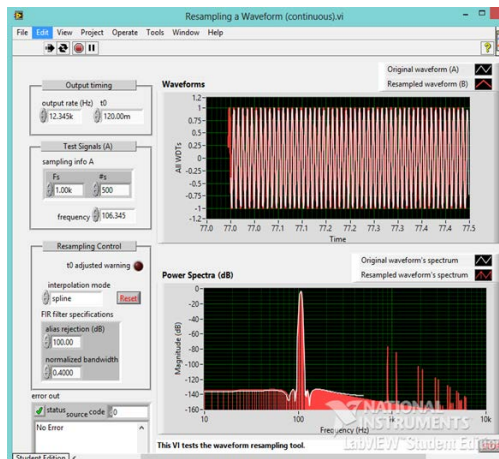Fig.11. Screenshot of a VI on Sampling of a Band Limited signal.

Fig.12. Example VI on resampling a waveform. Fig.13. Block diagram of a Waveform resampling VI

5.1   Sampling

A continuous-time (CT) signal can be processed by processing its samples through a discrete-time (DT) system. For reconstructing the CT signal from its DT samples without any error, the signal should be sampled at a sufficient rate that is determined by the sampling theorem.

5.2   Nyquist Sampling Theorem

If a signal is band limited and its samples are taken at sufficient rate than those samples uniquely specify the signal and the signal can be reconstructed from those samples. The condition in which this is possible is known as Nyquist sampling theorem. A real signal whose spectrum is band-limited to $D\,Hz\,[X\,(f)\,=\,0\,for\,|\,f\,|\,>\,D]$ can be reconstructed from its samples taken uniformly at a rate $fs\,>\,2D\,samples/sec$. We can say the minimum sampling frequency is $fs = 2D\,Hz$.

5.3   Aliasing

In reconstructing a signal from its samples, there is another practical difficulty. The sampling theorem was proved on the assumption that the signal $x\,(t)$ is band-limited. All practical signals are time limited, i.e., they are of a finite duration. A signal cannot be simultaneously time-limited and band-limited, but it can be simultaneously non-time and non-band limited. Clearly it can be said that all practical signals, which are necessarily time limited, are non-band limited, they have infinite bandwidth and the spectrum $X'(f)$ consists of overlapping cycles of $X(f)$ repeating every $fs\,Hz$ (the sampling frequency). Because of infinite bandwidth, the spectral overlap will always be present regardless of what ever may be the sampling rate chosen. Because of the overlapping tails, X'(f) has not complete information about X(f) and it is not possible, even theoretically to recover x(t) from the sampled signal x'(t). If the sampled signal is passed through an ideal lowpass filter, the output is not $X(f)$, but a version of $X(f)$ distorted as a result of some causes [10]:

a)   The loss of the tail of $X(f)$ beyond $|f\,|\,>\,fs/2\,\,Hz$.

b)   The reappearance of this tail inverted or folded onto the spectrum.

The spectra cross at frequency $fs/2 = 1/2T\,Hz$. This frequency is called the folding frequency. The spectrum folds onto itself at the folding frequency. A component of frequency $(fs/2) + fx$ shows up as or act like a component of lower frequency (fs/2)- fx in the reconstructed signal. Thus the components of frequencies above $fs/2$ reappear as components of frequencies below $fs/2$. This tail inversion is known as spectral folding or aliasing, which is shown in $Fig.\,5$. In this process of aliasing not only we are losing all the components of frequencies above $fs/2\,Hz$, but these very components reappear as lower frequency components. This reappearance destroys the integrity of the lower frequency components.

5.4   Sampling of Non-bandlimited Signal: *Anti-aliasing Filter*

Anti-aliasing filter is a filter, which is used before a signal sampler, to restrict the bandwidth of a signal to approximately satisfy the sampling theorem. The potential defectors are all the frequency

components beyond $fs/2\ Hz$. We should have to eliminate these components from $x(t)$ before sampling $x(t)$. As a result of this we lose only the components beyond the folding frequency $fs/2\ Hz$. These frequency components cannot reappear to corrupt the components with frequencies below the folding frequency. This suppression of higher frequencies can be accomplished by an ideal filter of bandwidth $fs/2\ Hz$. This filter is called the anti-aliasing filter. The anti-aliasing operation must be performed before the signal is sampled. The anti-aliasing filter, being an ideal filter is unrealizable. In practice, we use a steep cutoff filter, which leaves a sharply attenuated residual spectrum beyond the folding frequency $fs/2$.
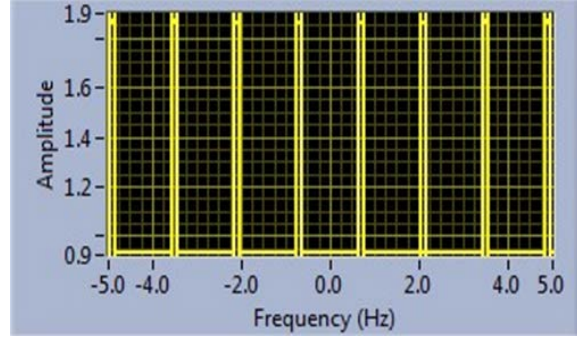

Fig.14 Aliased Signal

5.5 Signal Reconstruction Theory

The process of reconstructing a continuous time signal x(t) from its samples is known as interpolation. In the sampling theorem we saw that a signal x(t) band limited to D Hz can be reconstructed from its samples. This reconstruction is accomplished by passing the sampled signal through an ideal low pass filter of bandwidth D Hz. As seen in the above equation (3), the sampled signal contains a component and to recover x(t) or X(f), the sampled signal must be passed through an ideal lowpass filter having bandwidth D Hz and gain T.

The description of the process of reconstruction in the frequency domain is to find the DTFT of the discrete-time signal, change the variable F→f / fs , multiply by T and find the inverse CTFT. We find the DTFT from the samples using

$$x(F) = \sum_{n=-\infty}^{\infty} x[n]e^{-j2\pi Fn}$$

(1)

Here we are using the notations $X(f)$ for $CTFT$ and $X(F)$ for $DTFT$.

For sine function, the DTFT is illustrated in Fig.6. To isolate the function indexed by $k = 0$, we can multiply the DTFT by a rectangle function that is wide enough to include the $k = 0$ alias but not wide enough to include any other aliases. So the corner of the rectangle must be at a value of F which is greater than $Fm = fm/fs$, where fm is the highest frequency present in the signal $x(t)$ but less than $1 - Fm$ or more compactly, $Fm < Fc < 1 - Fm$. The isolated $k = 0$ replica is then

$$X_{F0}(F) = \text{rect}(\frac{F}{2Fc}) \sum_{n=-\infty}^{\infty} x[n]e^{-j2\Pi Fn}$$

(2)

Next we make the change of variable $F \rightarrow f/fs$ and multiply it by T

$$x(f) = TX_{F0}\left(\frac{f}{f_s}\right) = T\,\text{rect}\left(\left(\frac{f}{f_s}\right)/2F_c\right) \sum_{n=-\infty}^{\infty} x[n]e^{-j2\Pi fn/f_0}$$

(3)

Now we have to find out the inverse *CTFT*, which is

$$x(t) = F^{-1}\big(x(f)\big) = \int_{-\infty}^{\infty} T\text{rect}\left(\frac{f/f_s}{2F_c}\right) \sum_{n=-\infty}^{\infty} x[n]e^{-\frac{j2\Pi f_n}{f_{se}j2\pi f_s}}$$

Let $c = fc/fs$, then

$$x(t) = T \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n] \, \text{rect}\left(\frac{f}{2f_c}\right) e^{-j2\Pi f\left(f-\frac{n}{f_s}\right)df}$$

(4)

By exchanging the order of integration and summation we can recognize the integrand as a time shifted inverse *CTFT* of a frequency domain rectangle function.

$$x(t) = T \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \text{rect}\left(\frac{f}{2f_c}\right) e^{-j2\Pi f(t-nT)df}$$

(5)

Then $x[n] = x(nT)$,

$$x(t) = 2\left(\frac{f_c}{f_s}\right) \sum_{n=-\infty}^{\infty} x(nT)\text{sinc}\left(2f_c(t-nT)\right)$$

(6)

The reconstruction process consists of replacing each sample by a sine function, centered at the time of the sample and scaled by the sample value $x(nT)$ times $2fc/fs$ and adding all the functions so created. Suppose the signal is sampled at exactly Nyquist rate $fs = 2fm$, Then $fm = fs/2 = fs - fm$ and $Fm = 1/2 = 1 - Fm$.

The requirement $Fm < Fc < 1 - Fm$ cannot be met, in this case we must allow $Fc = Fm$, which means that $fc = fm = fs/2$. This will work till the signals spectrum does not have an impulse at fm. (If there is an impulse at fm, it will be aliased in the sampling process). In this limiting case, the interpolation is described by the simpler expression

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT)sinc\left(\frac{(t-nT)}{T}\right)$$

(7)

Interpolation consists of simply of multiplying each sine function by its corresponding sample value and then adding all the scaled and shifted sine functions. The interpolated value at any point is the sum of contributions from infinitely many weighted sine functions. There are many techniques that can be used to reconstruct a signal and the selection of the technique to be used is depends on what accuracy of reconstruction is required and how oversampled the signal is. Probably the simplest approximate reconstruction idea is to simply let the reconstruction always be the value of the most recent sample.

6 CONCLUSIONS

A general methodology for creating a web-based virtual communication laboratory has been presented. Based on this methodology, several experiments on sampling and signal reconstruction for students taking a course on Digital Signal Processing aspects of communication engineering has been developed in the department of Electrical and Information Engineering at Covenant University Nigeria. The laboratory requires only a web browser to access it, and incorporates schemes for reducing data traffic and authenticating users. It enables students to have a natural hands-on experience of using expensive instruments like the oscilloscope, spectrum analyzer and pulse generator on a one-to-one basis, and provides a solution for distant engineering education. The developed Virtual Laboratory system complements an earlier developed system on the iOS platform [1]. It is the aim of the Department to employ the method discussed in this paper for developing other Virtual laboratories for other programmes in the Department.

REFERENCES

[1]   J.O. Onaolapo, A.A. Atayero (2013), Design and Development of *CUvLab*: iOS-Based Virtual ECE Laboratory, Available at: http://bit.ly/at-vlab-13, Lambert Academic Publishing GmbH & Co. KG, Germany, ISBN: 978-3-659-45083-9, 133 pages.

[2]   Adams, Wendy K., et al. "A Study of Educational Simulations Part I-Engagement and Learning." *Journal of Interactive Learning Research* 19.3 (2008): 397-419.

[3]   Physical Sciences Resource Center available at: http://www.psrc-online.org/, accessed 2015.03.13

[4]   Stevens Institute of Technology, Remote Dynamical Systems Laboratory available at: [http://www.stevens.edu/remotelabs/], accessed 2015.03.13

[5]   Mercer University Online Interactive Chaotic Pendulum, available at: http://physics.mercer.edu/pendulum, accessed 2015.03.13

[6]   Curtin University available at: http://www.cage.curtin.edu.au/mechanical/info/vibrations, accessed: 2015.03.13

[7]   DSP Laboratory, IIT Kharagpur, available at: http://www.digital.iitkgp.ernet.in/dsp/

[8]   N. Ertugrul, Towards Virtual Laboratories: a Survey of LABVIEW-based Teaching/Learning Tools and Future Trends.

[9]   National Instruments, Computer-Based Measurement and Automation, Website, available at http://www.natinst.com (May 1999).

[10]   ZHOU YI, A LABVIEW-Based, Interactive Virtual Laboratory for Electronic Engineering Education.

[11]   National Instruments, LABVIEW tutorial on spectral analysis. (http://www.ni.com/tutorials).

[12]   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.

[13]   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.

[14]   National Instruments, LabVIEW User Manual, Austin, Texas (2003), pp.8-10.

[15]   National Instruments, LabVIEW Simulation Interface Toolkit User Guide, Austin, Texas (2003), p. 4.

[16]   B.P Lathi, "Signal Processing and Linear System" (Chapter-5, Sampling), 2nd edition Oxford University, 2008.

[17]   Michael J Roberts, "Fundamentals of Signals and Systems" (Chapter-14, Sampling and the Discrete Fourier Transform) Tata Mc-Hill 4th edition, 2009.

[18]   Alan V. Oppenheim, Alan S. Willsky with S Hamid Nawab, "Signals and Systems", Pearson Edition, 2003.

[19]   J.G. Proakis and D.G. Manolakis, "Digital Communication Techniques" 4th edition, Prentice Hall.

[20]   National Instruments, Computer-Based Measurement and Automation, Web site, available at http://www.natinst.com (May 1999).

[21]   LabVIEW Manual, National Instruments Corp., 1996.

[22]   N. Ertugrul, New era in engineering experiments: an integrated interactive teaching/ learning approach and real time visualizations, Int. J. Eng. Educ., 14, 5 (1998) pp. 344±355