

Improving Pedagogy in a Digital Signal Processing Course Using Computer-Aided Learning Systems

¹J.F. Opadiji

MIT Int. Science & Technology Initiative
MIT, Boston, Massachusetts, USA
jayeola@mit.edu

²A.T. Ajiboye, ³S.A. Olatunji

Department of Electrical Engineering,
University of Ilorin, Ilorin, Nigeria
²atajiboye@yahoo.co.uk

⁴A. Ayeni

Department of Telecommunication Science
University of Ilorin
Ilorin, Nigeria
aayeni@unilorin.edu.ng

ABSTRACT

The development and deployment of a computer-aided learning tool which serves as a learning aid for undergraduate course in digital signal processing is presented. We have carefully outlined the construction of the algorithm and the implementation process of the software with the aim of improving the level of understanding and assimilation of the concepts that make up the course. It also inspires software development at undergraduate level. Although the scope of this project was limited to fundamental principles of Digital Signal Processing, it was found to inspire software development among undergraduate students.

Keywords: Simulation; Computer-aided learning; Digital Signal Processing, CASdsp

1. INTRODUCTION

Digital Signal Processing is at the heart of modern development. Its advantages include: Smaller dimensions in designing, higher Reliability, better capability in complex processing and lower price of communication equipment. The demand on modern communication equipment is ever increasing. For example, the computers have brought with them, greatly expanded requirements on systems as well as on the already crowded radio frequency spectrum.

It is not surprising therefore that there have been considerable intents in Digital Signal Processing (DSP) to improve performance, testability and flexibility. Despite the promises of DSP in the area of promotion of technological development, especially in developing countries like Nigeria, the opportunities provided by adequate manpower development in this field of study has often eluded these nations. This problem stems from inadequate training received by professionals in this field. This project is aimed at providing a learning tool that will curb the advancement of this trend in a country like Nigeria where there is not only a dearth of manpower in this field, but also inadequate experimental tools to aid the understanding of students and trainee engineers [1, 2].

The problem we seek to alleviate is a complicated one that took its root from the facts that there are too few available resources for learning the concept of Digital

African Journal of Computing & ICT Reference Format:

J.F. opadiji, A.T. Ajiboye, S.A. Olatunji & A. Ayeni
(2011). Improving Pedagogy in a Digital Signal Processing
Course Using CAL Systems. Afr J. of Comp
& ICT. Vol 4, No. 3. P7-18

© African Journal of Computing & ICT September 2011
- ISSN 2006-1781

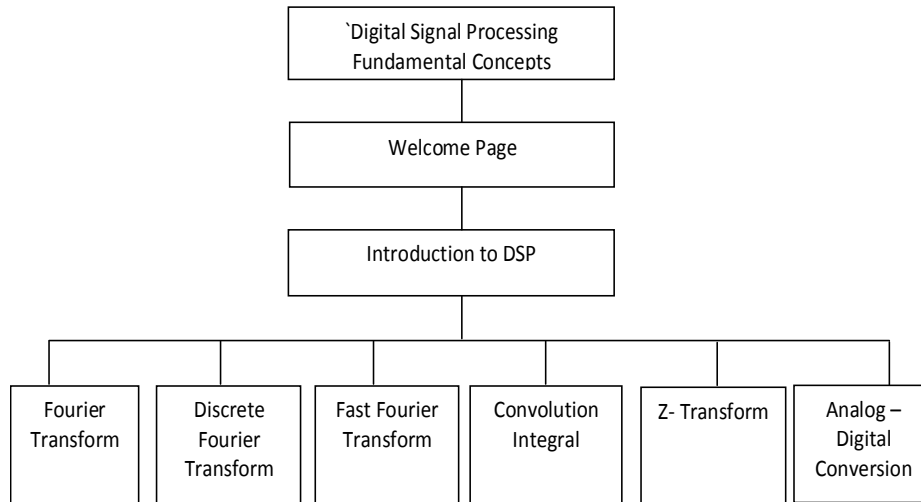


Figure 1: Modular Design of the Complete System

Signal Processing thoroughly in schools; non-availability of the microchips designed to Digital Signal Processing in most electrical college laboratories; scarcity of efficient software to handle the concepts; a high student to lecturer ratio in most of the higher institutions that offer this course

This project also aims at providing the opportunity for indigenous software development in Nigeria by motivating students to embrace software development activities as computer-aided systems similar to these are being implemented all over the world [3, 4, 5]

2. System Design and Algorithm Development

In order to develop a truly effective computer aided system for teaching and learning system (CASdsp), the various concepts in Digital Signal Processing (DSP) are broken down into modules in accordance with the top-down modular programming technique [6]. The main concept behind CASdsp is developed by writing program codes and building user interfaces in groups. Each of these groups or 'building blocks' which form the modules possess other functions which are carried out or executed within that particular group and sometimes infers relevant data from other groups within the system to carry out its commands. Each of these modules is created such that each contains a set of related, interconnected and sometimes interdependent instructions or commands. Once these sets of modules have been developed, the software package was then created by integrating these different modules such that each module can be accessed, upon user request by another module. The developed software has six main modules; each of these modules has sub-modules. The modular breakdown structure of the Computer Aided Learning System for DSP fundamental given the opportunity to read a brief introduction to the concept of Digital Signal Processing under the caption "Introduction".

concepts is shown in Fig 1. Each module was then further broken down into three segments as explained below:

2.1 The Input Interface

The input interfaces were designed to receive system parameters as designed for the algorithms developed. These interfaces were designed specially to be very sensitive to errors such that the user will be guided as to which form, type and nature of data are applicable to the concept being handled.

2.2 The Processing Engine

This sub-module is dedicated to working out the solution to the problem created using the input parameters and carrying out specific operations as indicated in the algorithm developed for that module. The results emanating from the calculations and operations performed are then sent to the output interface for display.

2.3 The Output Interface

Results obtained from the operations of the processing engine are sent here for display. This sub-module is closely linked with the input interface displaying parameters entered by the user as input, the results generated after processing and then the graphical representations of the output.

3. SYSTEM IMPLEMENTATION

The implementation of the design was done using the Microsoft Visual BASIC Integrated Development Environment to make it user user-friendly [7], therefore, the CASdsp uses an event-driven, object-oriented programming language. Shown in Figure 2 is the main menu of the developed software and sample results of some of its modules: On this form, the user (the student) is

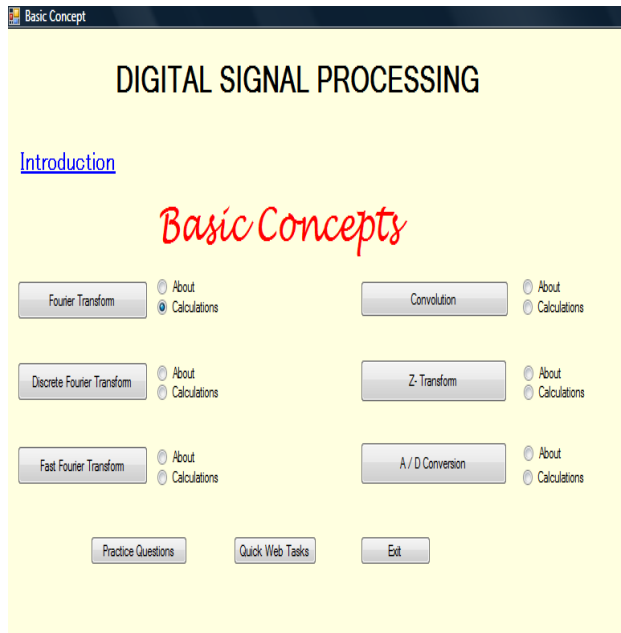


Figure 2: Main Menu of the Developed package

Then one can proceed to know about each of the fundamental concepts by clicking on the “about” radio button and then clicking on the desired concept. The program is very user – friendly and designed specifically with the undergraduate curriculum in mind such that important issues under each of the concepts are displayed at the request of the user, ranging from origin of the concept to the applications of concepts in real life problems as well as examples and practice questions on the concept selected.

All these are put in place in other to give the student sound background knowledge of the concept. Then, to practice what has been read, another interface was developed where the user can apply some of the principles learnt from the theoretical knowledge

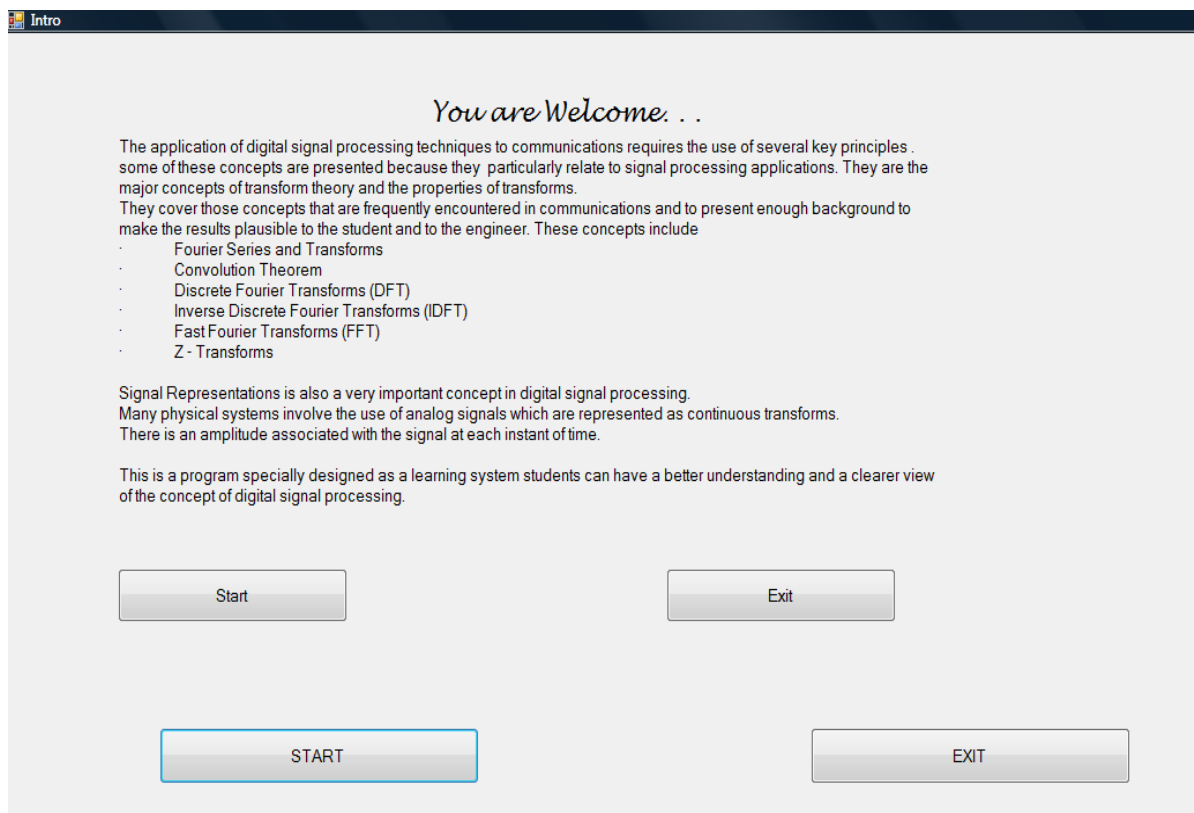


Figure 3: Introductory Page to Digital Signal Processing Concepts

Then one can proceed to know about each of the fundamental concepts by clicking on the “about” radio button and then clicking on the desired concept. The program is very user – friendly and designed specifically with the undergraduate curriculum in mind such that important issues under each of the concepts are displayed at the request of the user, ranging from origin of the concept to the applications of concepts in real life problems as well as examples and practice questions on the concept selected. All these are put in place in order to give the student sound background knowledge of the concept. Then, to practice what has been read, another interface was developed where the user can apply some of the principles learnt from the theoretical knowledge.

3.1 Fourier Transform Module

Fourier Transforms are primarily used to represent finite-length analog signals [8, 9, 10], although they can be used to represent some continuous signals such as $\sin(\omega t)$ and $e^{j\omega t}$.

On clicking on the “Calculation” radio button and clicking Fourier Transform button, the main menu window is unloaded and message box appears prompting the user to select the time format. The form shown in Figure 4 for example, is the window which comes up when “Continuous Time” is chosen as the time format “Exponential function” has been selected as well on this form as the type of input. On selection of the type of input, a textbox appears which enables users to enter the amplitude and period of the function selected. The software calculates the Fourier transform and displays the output in frequency domain when the “Transform” button is clicked. Also a graphical representation of the input signal coming in time domain and the output signal graph coming in frequency domain is displayed when the “Plot Graph” button is clicked. This enables the user to view the graphical representation of the type of input as well as the characteristics of the function that was entered as well as the properties of the frequency function coming as the output after being transformed using Fourier Technique

All these can be seen in Figure 4. Other types of input that the user can access are Fourier series and Complex Fourier Coefficient which also give, as output, the function in frequency domain as well as the graphical plot of the input and output functions. All these can also be done in discrete format if the user selects the “Discrete Time” format when prompted.

3.2 Discrete Fourier Transform Module

The Discrete Fourier Transform is the parallel of the continuous Fourier transform for sampled data system [8]. The input parameter window enables users to enter the various time – domain functions to be transformed into frequency domain functions. The software is designed to give allowance of, as much as, four different input signals $x(n)$ and their ranges at a time. The number of points to be used for the computation of the DFT is also

entered by the user. The software also gives the user the opportunity to view the graphical representation of each of the input signal independently as the functions are typed in even before the individual functions are transformed. This is shown in Figure 5. Then, on clicking “Transform”, the output, which is the frequency domain equivalent of the function typed in, as input, is displayed and a graphical representation of the output is displayed alongside. This is shown in the Figure 6.

3.3 Fast Fourier Transform Module

Fast Fourier Transform algorithms (FFTs) collectively form an enormous number of fast algorithms for the computation of the DFT [8, 9]. The most popular types which are the ones used in this program are the Radix -2 algorithm and the Radix -4 algorithm which both come with decimation in time and decimation in frequency.

On clicking on the “Fast Fourier Transform” button on the Basic Concepts Menu window, after indicating “Calculation” on the radio button, a message box asking which FFT format to be used pops up. On this message box are two options: Radix – Two and Radix – Four. The form shown above is the result of clicking on Radix – Two and on this form input parameter window for Radix-Two FFT calculation. Here, the user is given the allowance to input as many as four separate and independent equations representing functions in the time domain $x(n)$ with $n = 0, 1, \dots, (N/2) - 1$. Also, the period for the time-domain functions N is as well required as an input. A privileged opportunity is then given to the user to view individually and independently the graphical representation of each of the input functions entered in line with the period (N) entered. All these are designed by the programmer to make the CALS as user-friendly as possible. This gives the user an idea of the characteristics of the input signals. This is shown in Figure 7

The form shown in Figure 8 displays the result obtained when the ‘Transform’ button is clicked by the user to get result of the transformation of the input, time-domain functions to frequency – domain functions as the output. The result is shown in the space provided for the output just below the input and a butterfly representation showing how the result was obtained is displayed immediately just alongside the result as shown in the Figure 8. If the Radix Four FFT algorithm is selected, then the Radix Four FFT algorithm window as shown in Figure 9 pops up. In this window, the user is allowed to enter 16 various input parameters that come as time domain functions $x(n)$. the number of FFT points is to be entered by the user. The result is that, as shown in Figure 8, where the program computes the FFT of the functions entered and displays the results individually just to make the user see in a well defined form the result of each of the individual functions $g(1)$ to $g(16)$. Afterwards, a butterfly representation to explain how each of the results were computed is displayed when the user clicks on “Details”. This is shown in Figure 4.

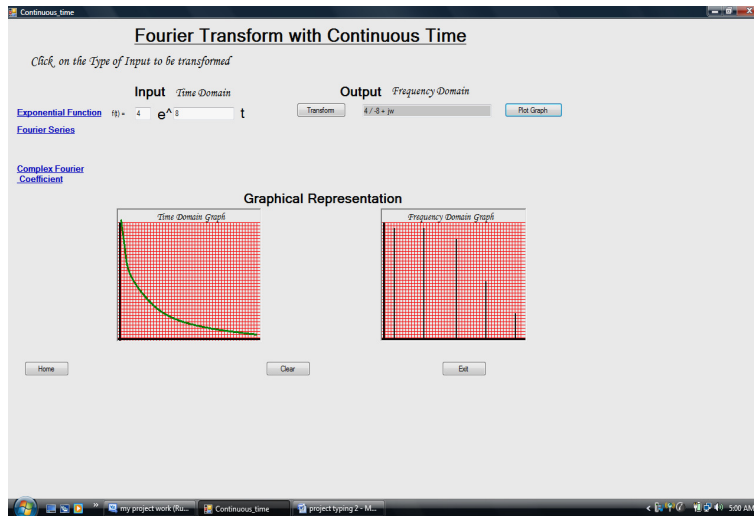


Figure 4: Sample result of Fourier Transform Module

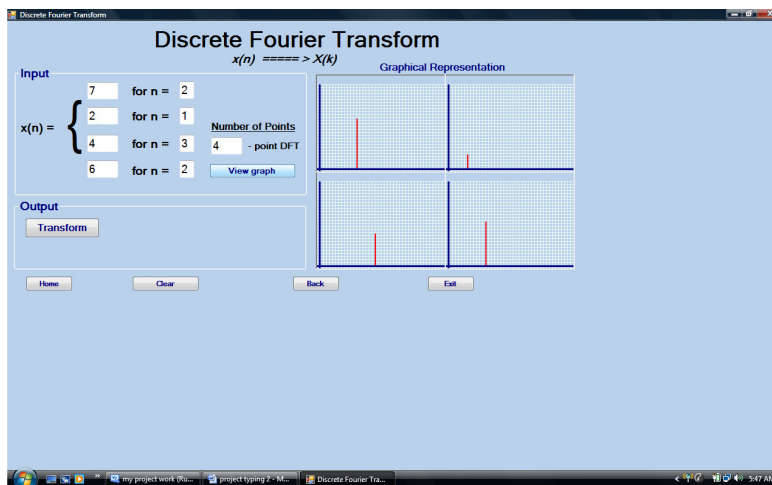


Figure 5: Input Parameter Window for Discrete Fourier Transform

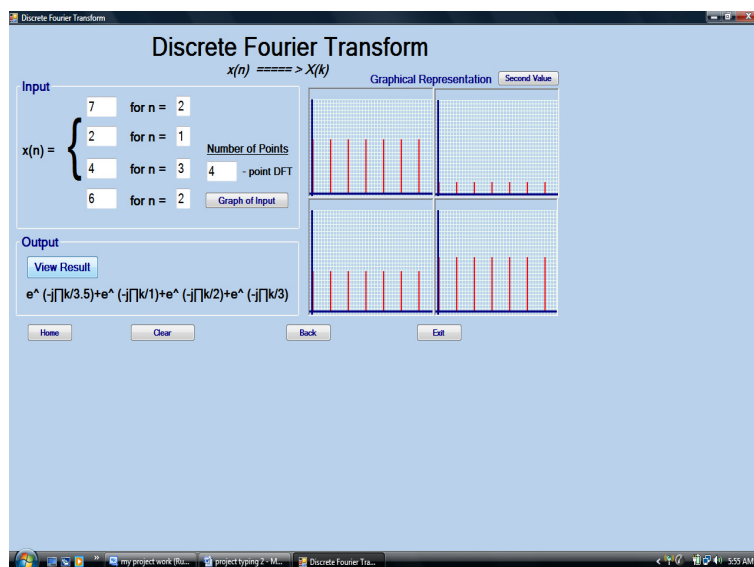


Figure 6: Sample result of the Discrete Fourier Transform Module

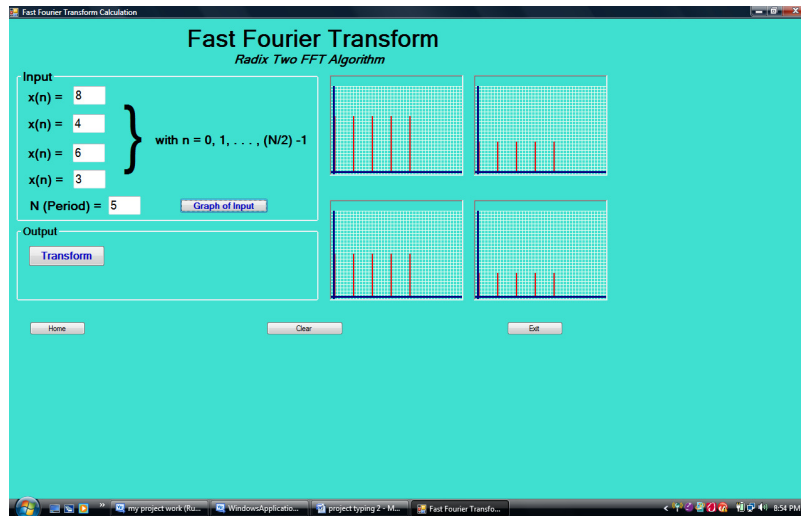


Figure 7: Input Parameter Window for Fast Fourier Transform

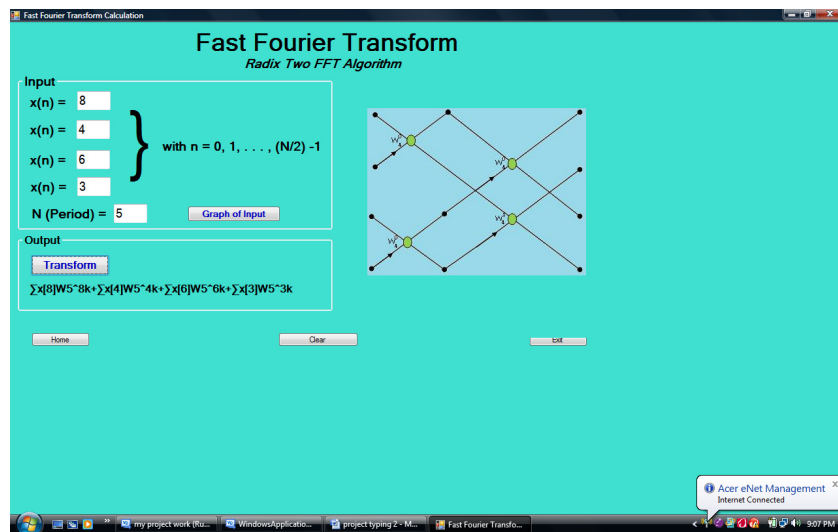


Figure 8: Sample result of the Radix two Fast Fourier Transform Module

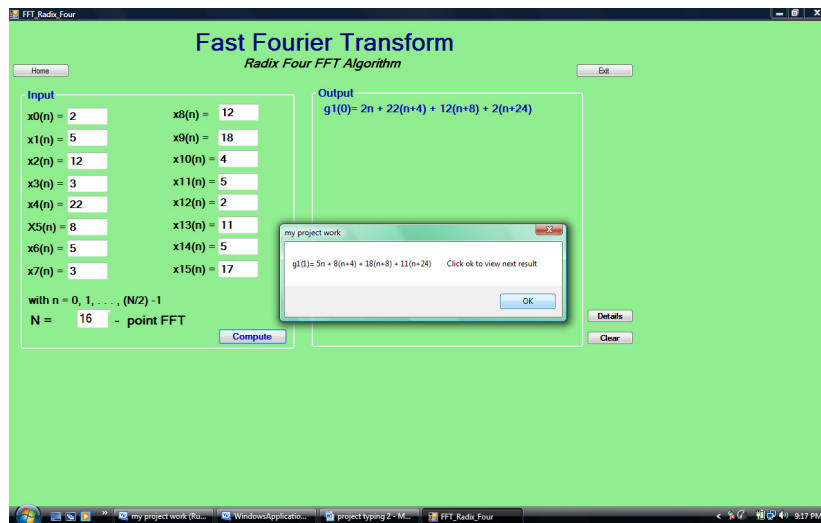


Figure 9: Radix Four FFT algorithm window

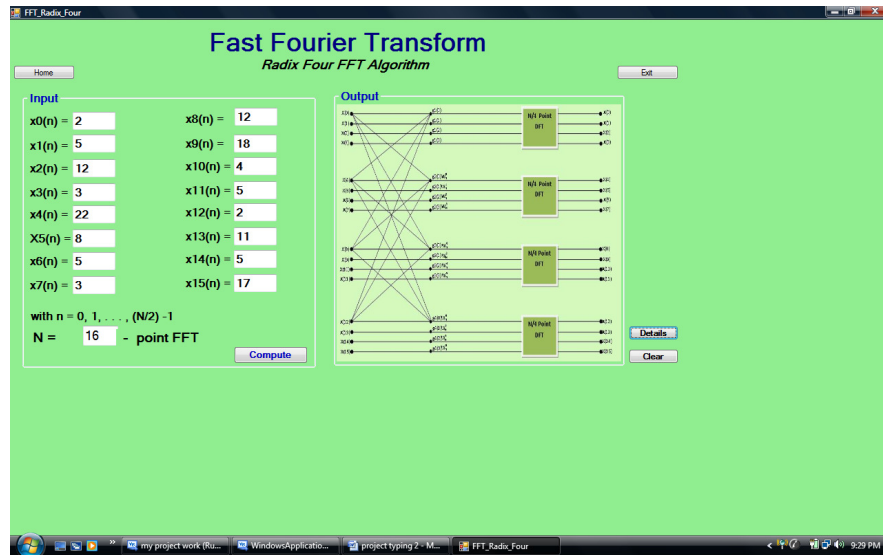


Figure 10: Sample result of the Radix Four FFT module

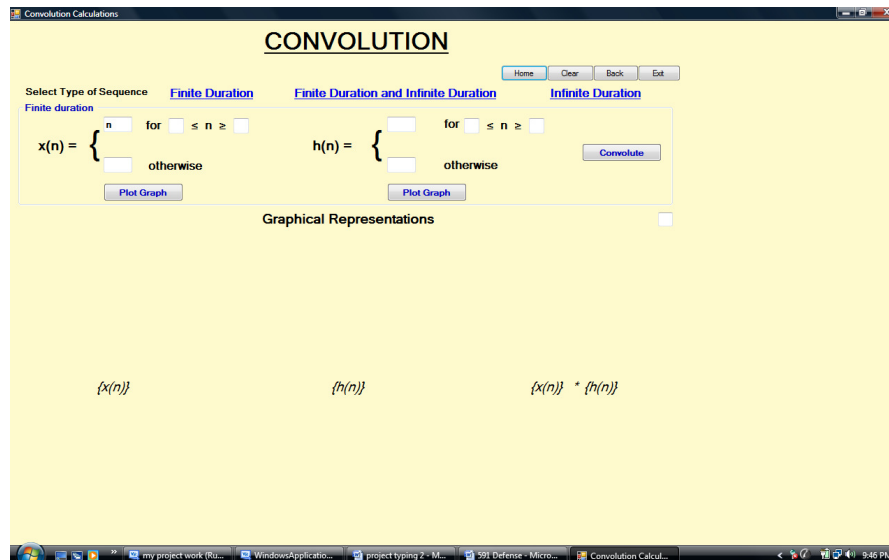


Figure 11: Input parameter window for Convolution Integral

3.4 Convolution Module

Convolution helps to give the result of the intersection of the region of convergence of analog input [7]. This comprises of the concept definition and explanation part as well as the module where the knowledge gained can be applied in form of calculations. Clicking on “Convolution”, after selecting “Calculation” on the radio button, loads the convolution calculation window where the user can first select the type of sequence desired. We can view the individual plot of the input parameters, in form of graphs, even before convolution takes place.

The result of the convolution as well as the graphical representation of the region of intersection is also shown in Figure 12.

The result shown in Figure 12 is a sample of the output obtained when ‘infinite duration’ is selected as the type of sequence and the input parameters representing the analog inputs are entered in form of $x(n)$ and $h(n)$. Individual graphs representing the domain of $x(n)$ and $h(n)$ respectively is displayed immediately the “Plot Graph” button is clicked by the user. The result, as well as the result of the intersection, is displayed when the user clicks on “Convolution” button

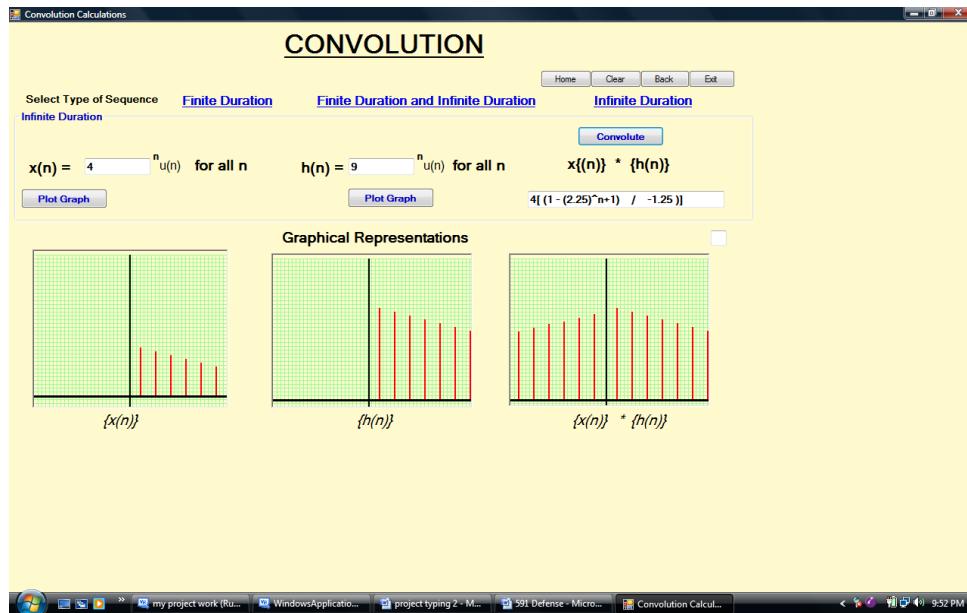


Figure 12: Sample result of the Convolution Integral module

3.5 Z- Transform Module

The Z –transform is only defined for the regions of the complex plane in which the summation on the right converges [6]. Very often, the signals start at $n=0$, that is, they are nonzero only for $n \geq 0$. If the user clicks on 'Z-Transforms' after checking the radio button for 'Calculation', the Z-transforms window for calculation pops up, which gives the user the opportunity to select different types of input to be

transformed such as discrete unit pulse, step function, unit pulse shifts, exponential function and trigonometric function. The student is thus given the opportunity to select one type of function at a time. Then, the user enters the function applicable to the type he has chosen. On clicking the 'transform' button, the result of the transformation is displayed as well as a graphical representation of the input.

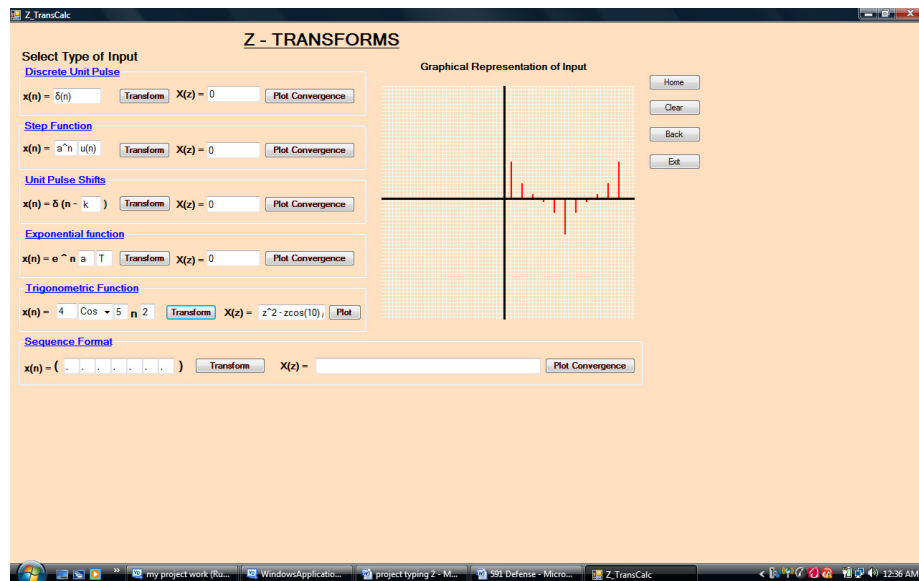


Figure 13: Input Parameter Window for Z-Transform

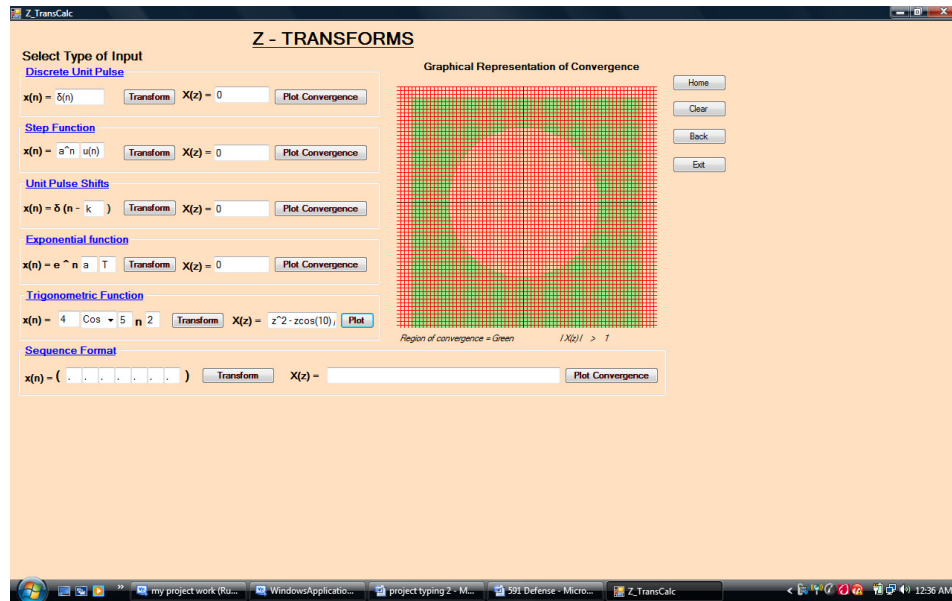


Figure 14: Sample result of Z-Transform displaying region of convergence

3.6 Analog – Digital Conversion Module

Basic Processes involved in analog to digital signal conversion are, anti-aliasing, sampling, quantizing, encoding [5, 6, 7]. The introductory section is where the user can read about the basic processes involved in

the conversion of analog signals to digital signals. On clicking the corresponding button for each of the concepts underlined, more explanation on each of these concepts are displayed for the user to read and learn. A sample is shown in Figure 15.

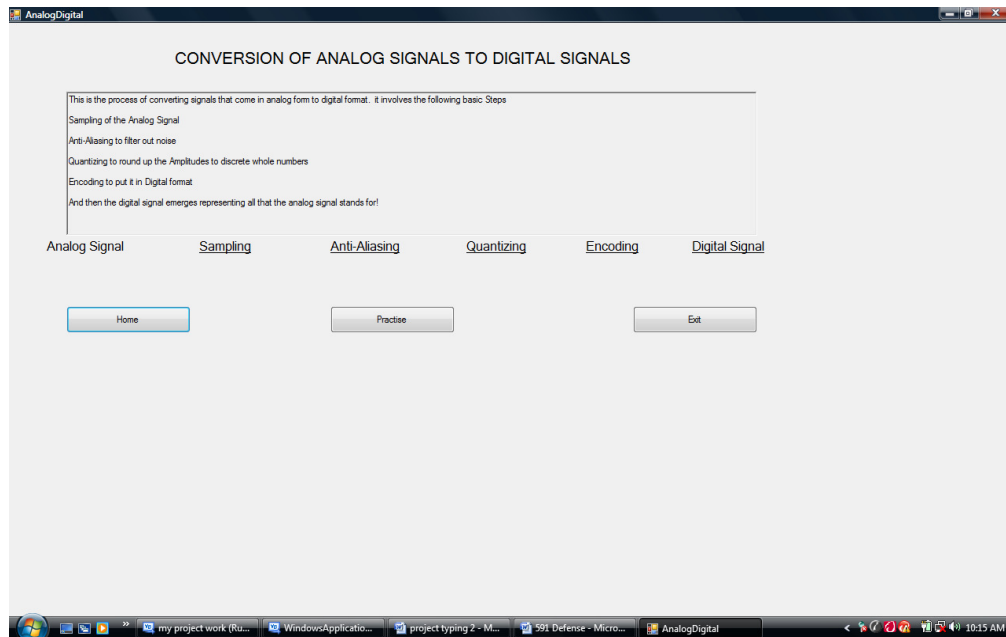


Figure 15: Introduction Module for Analog to Digital Conversion

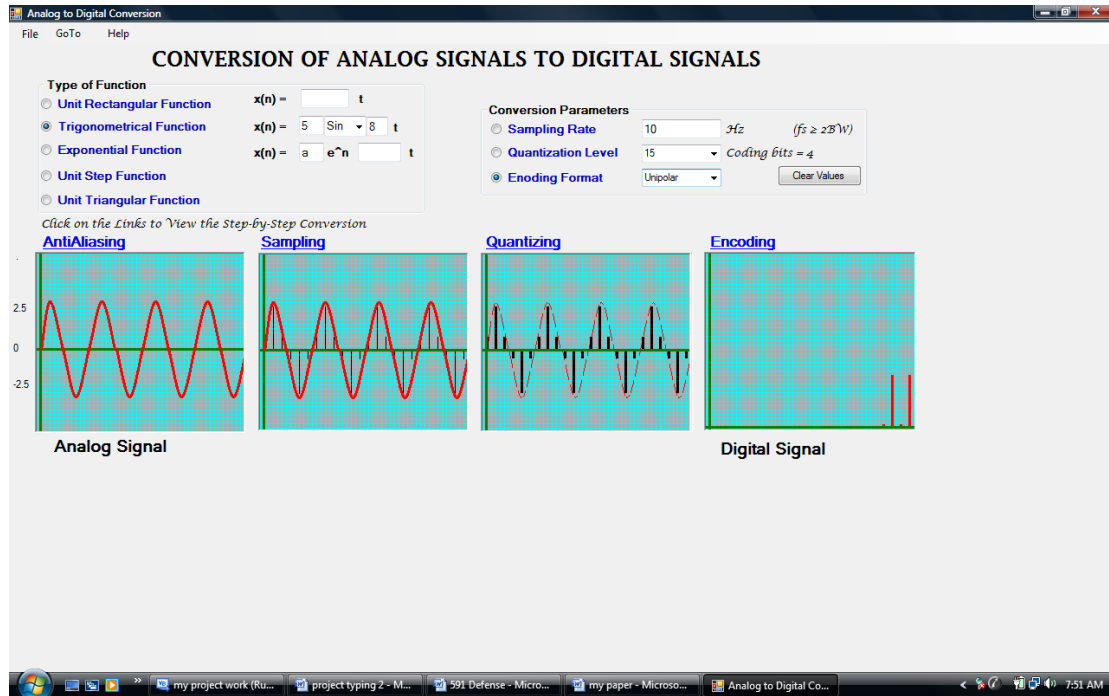


Figure 16: Sample result of Analog to Digital Conversion Module

When the user is ready to implement the knowledge learnt, he clicks on 'A-D Conversion' after marking the 'calculation' radio button. The window for Analog to Digital Conversion pops up. Here, the user can first select the type of function: ranging from unit rectangular function, trigonometric function, exponential function, even to unit step function, among others.

Then he enters the input (analog) signal, the sampling rate, to determine the amplitude at each point of the continuous signal and then enter the quantization level. Once the quantization level is selected, the bit resolution is displayed as applicable to the selected level. The encoding format, whether bipolar or unipolar coding, is then selected. Thereafter, the student clicks the various stages of the conversion to see what graphical representation at each stage of the conversion looks like.

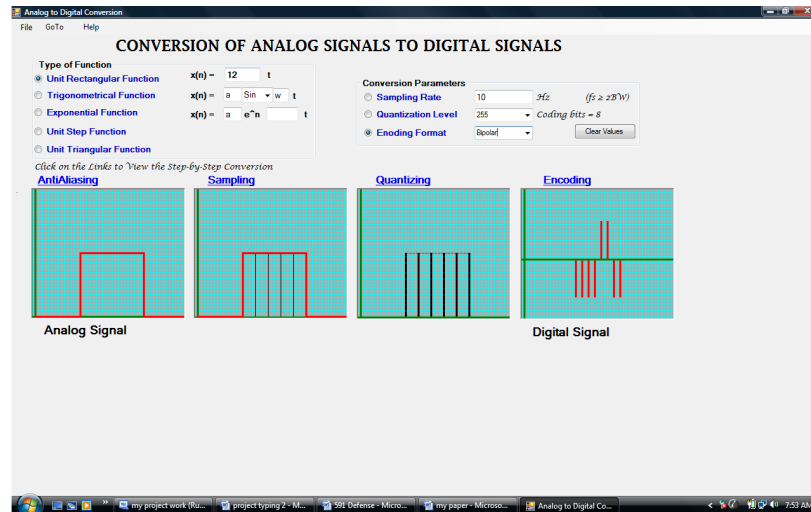


Figure 17: Sample result of Analog to Digital Conversion using unit rectangular function

4. DISCUSSION

After the successful completion of the design implementation stage, the software was compiled into an executable file and currently runs as a stand-alone application pending the time it will be deployed on some systems in the Department of Electrical Engineering, University of Ilorin. The software is presently undergoing scrutiny and criticism from users which will enable it to be improved based on the response and reaction of the users after their interaction with the software. Modifications can be done to meet up with users' expectation of the program. The deployment was done on systems running Microsoft Vista and Microsoft Windows XP.

5. CONCLUSION

Obviously, our tertiary learning institutions require better training and learning facilities to cope with the fast rate of technological advancement in this age and time. Since computers can now be easily afforded by these institutions, it is pertinent that we seize the opportunity to develop computer aided learning systems which will aid better and faster learning as well as efficient teaching in the classroom. The pace of learning can be controlled and improved and this also means that students can make better and more intelligent choices in what and how to learn, skipping unnecessary items or doing remedial work in difficult concepts, all with aim of understanding better and faster. Such control makes students feel more competent in their learning.

As regards this project, there is still more work to be done in the other areas and aspects that make up the concept of Digital Signal processing and there are already interesting developments in the pipeline. The program would be upgraded with time to suit the needs, desires and convenience of the user as well as to make a more efficient learning package and an apt teaching tool. This software is adjudged to have solved the problem of enabling the students grasp better, the basic concepts of digital signal processing thus satisfying the objective.

Just like every other software developed, it has its limitations part of which is the fact that the scope is limited to just the fundamental principles in DSP and it is designed for training in institutions of learning not as a replacement for microcontrollers used for digital signal processing in equipments and appliances. Experts can however make use of it to further train interns and apprentice who have at least a good background in the field.

REFERENCES

- [1] M Nwadiani and N.E. Apotu (2002): *Academic Staff Turnover in Nigerian Universities*; Education, Vol. 123.
- [2] B. Oni (1999): *The Nigerian University Today and the Challenges of the Twenty-first Century*; Monograph, No. 60. Institute for World Economics and International Management, University of Bremen, Bremen, Germany.
- [3] M.W. Goldberg (1997): CALOS: first results from an experiment in computer-aided learning for operating systems, Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education, NY, USA,
- [4] P.S. Goodman (1998): Computer-Aided Systems and Communities: Mechanisms for organizational learning in distributed environments; Management Information System Quarterly, University of Minnesota, (22) 4.
- [5] J.N. Hudson (2004): Computer-aided learning in the real world of medical education: does the quality of interaction with the computer affect student learning?; Medical Education, (38) 8, pp 887 – 895.
- [6] A.V. Aho, J.E. Hopcroft and J.D. Ullman (1974): *The Design and Analysis of Computer Algorithms*; Addison-Wesley Publishing Company, Essex, England.
- [7] G. Brassard and P. Bratley (1996): *Fundamentals of Algorithms*; Prentice Hall, New Jersey, USA.
- [8] S.K. Mitra (2005): *Digital Signal Processing: A Computer-based Approach* (Third Edition); McGraw-Hill Series in Electrical and Computer Engineering; McGraw-Hill Higher Education.
- [9] J.G. Proakis, D.G. Manolakis (1996): *Digital signal processing: Principles, Algorithms and Applications* (Third edition); Pearson Prentice Hall, New Jersey
- [10] P.S.R. Diniz, E.A.B. Da Silva, S.L. Netto (2002): *Digital signal processing: system analysis and design*; Cambridge University Press, U.K.