Jayeola F. Opadiji
NJTD (2010) 7(1) 48- 57
ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

SDEVELOPMENT OF A COMPUTER-AIDED LEARNING SYSTEM FOR GRAPHICALANALYSIS OF CONTINUOUS-TIME CONTROL SYSTEMS

JAYEOLA F. OPADIJI
Department of Electrical Engineering, University of Ilorin, Ilorin, Nigeriae-E-mailjopadiji@unilorin.edu.ng

ABSTRACT
We present the development and deployment process of a computer-aided learning tool which serves as a training aid for undergraduate control engineering courses. We show the process of algorithm construction and implementation of the software which is also aimed at teaching software development at undergraduate level. The scope of this project is limited to graphical analysis of continuous-time control systems.

Keywords: Simulation, Computer-aided learning, Continuous-time control system

1.0 INTRODUCTION

In recent years, there has been a shortfall in the number of students offering to option in the field of Control Engineering at undergraduate level in Nigerian universities. In fact, this has been so dramatic that, some universities no longer offer the discipline as an option in the Electrical and Electronic departments. When investigated, it was found out that, a number of reasons have contributed to this, but prominent among these are the fact that, there are not enough teaching and laboratory materials to assist in the disseminating of knowledge in this field. Also, there has been a great drop in teaching personnel who specialize in Control Engineering (Nwadiani this field. Also, there has been a great drop in teaching personnel who specialize in Control Engineering (Nwadiani and Apotu, 2002; Oni, 1999). It is against this backdrop that this project was initiated; to provide a software teaching aid for the introduction of students to the field of control engineering.However, it was discovered that software like this will not only serve as a teaching aid for control engineering students but also aid in teaching engineering students some fundamental principles of computer programming. The factors mentioned above served as tools in designing the scope and objectives of this work.

Control Engineering is a field of study, which cuts across every area of human existence. It involves the use of available resources to cause a system to behave in a predetermined manner based on information obtained about the system. The evolution of Control Engineering practice has brought with it, the introduction of complex analytical tools for the purpose of modelling and predicting systems whose behaviour would of course have to be understood before they can be controlled.

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

In developing any system, time always happens to be a major constraint, therefore, the use of complex graphical and mathematical tools, which require long hours of manual implementation may not be appropriate. Before any accurate plot can be done manually, it is required that a table of values must be gotten and this could be very tedious especially for high-order systems. Furthermore, the process of plotting itself is energy sapping if the points are many. Using machines however, is not automatic, as they have to be instructed in what to do in the form of programmes written to perform specific tasks. Algorithms have to be developed for any of the analytical tools before they can be implemented.

Thus, this is the first hurdle to be crossed in the development of appropriate software. The implementation of these algorithms is also a task to be accomplished, paying full attention to software development principles (Aho, Hopcroft and Ullman, 1974; Brassard and Bratley, 1996)

The scope of the project work covers graphical analytical tools used in classical control systems:Time response graph

(i.) Root Locus

(ii.)Nyquist plot

(Iii.)Bode plot

(iv.)Nichols plot.

The above tools were implemented in a computer software environment for the analysis of continuous-time control systems.

2.0 PROJECT OBJECTIVE

There were two main objectives of this software development project and these are enumerated below:

(i.) The development of suitable algorithms for analysis of systems defined in the scope of the project work.

(ii.) The implementation of the developed algorithms using an appropriate programming language that will aid in teaching software development.

These objectives were set with due consideration of the target users of the computer programs.

3.0 System Design and Algorithm Development

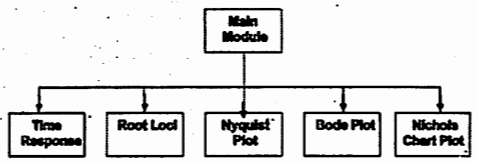


Figure 1: Modular Design of the Complete System

A modular approach was employed at the design stage of the software (Aptech, 1999). The first step taken was to break the whole system down into modules and each module was then developed separately before final integration to form a whole system as shown in the modular plan of the whole system. Figure 1 shows a modular plan of the whole system. Each module was further broken down into three segments as explained below:

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

3.1 The Input Interface:

The input interfaces were designed to receive system parameters as designed for in the algorithms developed. These interfaces were designed to trap errors in data entry so that user will be guided as to which data are valid for the different system parameters used as inputs.

3.2 The Processing Engine:

This sub-module is responsible for carrying out specified operations on the input parameters as laid out in the developed algorithm for that module. The results of these operations are sent to the output interface for display.

3.3 The Output Interface:

Each output interface is made up of the graph section as well as the performance parameter section. Results gotten from the processing sub-module are displayed in this interface.

4.0 ALGORITHMDEVELOPMENT:

Algorithms were designed for the five modules of the project. In developing the algorithms, adequate attention was paid to computability of each of the steps since the computer requires that each step must not be ambiguous. Moreover, the algorithms are meant to be used for the purpose of teaching programming to students.

4.1 Time Response Algorithm

The algorithm is based on a typical second-order system with the characteristic equation:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \qquad (1)$$

1. Get coefficients of the second order polynomial
2. Get maximum response time t
3. Get graph resolution r
4. Dimension y(t/r)
5. Calculate undamped natural frequency $_n$ from:

$$\omega_n = \sqrt{c^{\omega}} \qquad (2)$$

6. Calculate damping coefficient from:

$$\zeta = b / 2\sqrt{c} \qquad (3)$$

7. If $0 \le \zeta \le 1$ proceed else terminate programme
8. Generate plot data for i = 0 to T using:

$$y(i) = 1 - \frac{e^{-\zeta\omega_n i}}{\sqrt{1-\zeta^2}} Sin(\omega_n \sqrt{1-\zeta^2} i + \phi) \qquad (4)$$

9. Plot graph using the plot data generated
10. Calculate system performance parameters
11. alpha = damping constant= $\zeta\omega_N$
12. beta = damped natural Frequency = $\omega_n\sqrt{(1-\zeta^2)}$
13. rise time = 1.8 / nf
14. peak time = 3.142 / beta
15. Settling time = 4 / alpha
16. peak overshoot = Exp((-3.142 * alpha/nf)/(beta / nf)):

4.2 Root Locus Algorithm

The open loop transfer function is given as:

$$\frac{KD(s)}{N(s)} = 0 \qquad (5)$$

Algorithm is based on root-locus sketch rules:

1. The number of loci is equal to the number of poles of the open-loop transfer function
2. For values of K greater than 0, points on the root locus lie to the left of odd number of poles and zeros of the open-loop transfer function.
3. If the number of poles is greater than the number of zeros, the difference is equal to the number of loci that approach asymptotes which have a center on the real axis given by:

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

$$\alpha = \frac{(2l+1)180^0}{n-m}(l = 0,1,2,...) \quad (6)$$

And the angle between the asymptotes is given

$$\sigma_c = \frac{\sum_{l=1}^{n} p_l - \sum_{l=1}^{m} z_l}{n-m} \quad (7)$$

5.If a pole and a zero are next to each other on the abscissa, draw a line from the pole to join the zero.

6.If two poles or two zeros are next to each other on the real axis, then there is a breakaway or break-in point the value of which is gotten as defined below:

$$\sum_{l=1}^{n} \frac{1}{(\sigma_b + p_l)} = \sum_{l=1}^{m} \frac{1}{(\sigma_b + z_l)} \quad (8)$$

7. Where p_l and z_l are poles and zeros on GH(s) and $_b$ is the breakaway or break-in point.

8.This results in an (n+m-1) order polynomial, one of the solutions of which gives the desired point on the abscissa.

9.The angle of departure and of a complex conjugate pole and angle of arrival of a complex conjugate zero of GH(s) are stated respectively as:

$$\theta_D = 180^0 + \angle GH' \quad (9a)$$

$$\theta_A = 180^0 - \angle GH' \quad (9b)$$

10. Where $\angle GH'$ is the phase angle of the open-loop transfer function obtained at a point close to the complex conjugate pole. The game margin of the system is defined as:

$$GM = \frac{K_{SB}}{K_D} \quad (10)$$

12. 1Where K_{SB} = value of K at the stability boundary

13. K_D = Design value of K

14.The phase margin is given as:

$$\phi_{PM} = 180^0 - \angle GH(\omega_1) \quad (11)$$

15.Where $_1$ = value of at the stability boundary for which |GH| = 1 for the design value of K

16. Draw sketch lines using the calculated parameters and axioms given above

4.3 NYQUIST PLOT ALGORITHM

The Nyquist Plot is a mapping of the Nyquist path from the s-plane to the j-plane

Algorithm for Nyquist path:

1. Get zeros and poles

2. Draw a large semi-circle with centre at the origin of the j axis of this s-plane using:

$$f(x,y) = \lim_{R\to\infty} Re^{j\theta} (-90 \le \theta \le +90) \quad (12)$$

3. Evade all points of singularities on the ordinate with a semi-circle using

$$f(x,y) = \lim_{\rho\to\infty} \rho e^{j\theta} (-90 \le \theta \le +90) \quad (13)$$

Nyquist Plot:

1.Do the Nyquist plot using the following mapping from the Nyquist path (s-plane) into the j - plane: ω

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

2.The limit of the straight line portion on the s-plane is given by

$$s = \pm j\omega \quad (0 < \omega < \omega_0) \qquad (14)$$

3.Where ω_0 is a point of singularity

4.The limits of the small semi-circle on the ordinate is given by

$$s = \lim_{\rho \to 0} \rho e^{j\theta} \quad (-90^0 < \theta < +90^0) \qquad (15a)$$

5.The limits of the large semi-circle is given by

$$s = \lim_{R \to 0} Re^{j\theta} \quad (-90^0 < \theta < +90^0) \qquad (15b)$$

6.The limits of the poles on the j - axis is

$$s = \lim_{\rho \to 0} (\pm j\omega + \rho e^{j\theta}) \quad (-90^0 < \theta < +90^0) \qquad (16)$$

4.4 BODE PLOT ALGORITHM

The Bode form for the open-loop transfer function used is:

$$\frac{K(1 + j\omega/z_1)(1 + j\omega/z_2)\ldots(1 + j\omega/z_m)}{j\omega^l(1 + j\omega/p_1)(1 + j\omega/p_2)\ldots(1 + j\omega/p_m)} \qquad (17)$$

Where z's and p's are zeros and poles respectively and l is a nonnegative integer

1.Get number poles and zeros
2.Get maximum value of ω(max) and resolution r
3.Initialize i = 1
4.Calculate the dB value for each term in GH(ω) for ω = i
5.Add all the terms to get the dB value for GH(i)
6.Find the phase angle for each term in GH(ω) for ω = i
7.Add all the terms to obtain the phase angle of GH(i)
8. Find the logarithm valu of ω = I
9. Plot point in step 5 against point in step 8
10. Plot point in step 7 against point in step 8
11. Increment i by r
12.If I is less than max go to step 4
13. Terminate run

4.5 NICHOLS CHART PLOT ALGORITHM

The plot is done on a chart made up of constant dB magnitude circle and constant phase angles circles. The constant dB-magnitude circles are gotten from the relation:

$$|GH(\omega)|^2 + \frac{2M^2}{M^2 - 1}|GH(\omega)|Cos\theta + \frac{M^2}{M^2 - 1} = 0 \qquad (18)$$

The constant phase angle circles are gotten from:

$$|G(\omega)| + Cos\theta - \frac{1}{N}Sin\theta \qquad (19)$$

1. Get number poles and zeros
2. Get maximum value of (max) and resolution r
3. Initialize i = 1
4. Calculate the dB value for each term in GH(ω) for = i
5.Add all the terms to get the dB value for GH(i)
6. Find the phase angle for each term in GH(ω) for = i
7. Add all the terms to obtain the phase angle of GH(i)
8.Plot point in step 5 against point in step
9. Increment i by r
10. If i is less than max go to step 4
11. Terminate run

5.0 SYSTEM IMPLEMENTATION

The implementation of the design was done using the Microsoft Visual BASIC 6.0. The choice of this software was necessitated by the need to demonstrate the process of Rapid Application Development (RAD) and also take advantage of its Graphic User Interface (GUI) to produce a user-friendly environment for the software that is being developed. Shown below is the main menu of the developed software and sample results of two of its modules.

5.1 TIME RESPONSE MODULE

On clicking on the "Time Response" button on the main menu window of figure 2, the main menu window is unloaded and the time response input parameter window pops up as shown in figure 3.

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

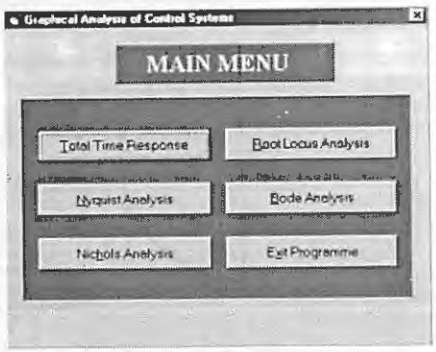


Figure 2: Main Menu of the Developed Package

The Input parameter window enables users to enter the coefficients of the characteristic equation for any second order system as well as the time interval for within which the user wishes to monitor the system response. The software calculates the damping ratio and the undamped natural frequency for the system. If the damping ratio is between 0 and 0.67, the "Plot Graph" button is enabled. The input parameter window is hidden when the user clicks on the "Plot Graph" button and the graph window for the time response module is loaded as shown in Figure 4.

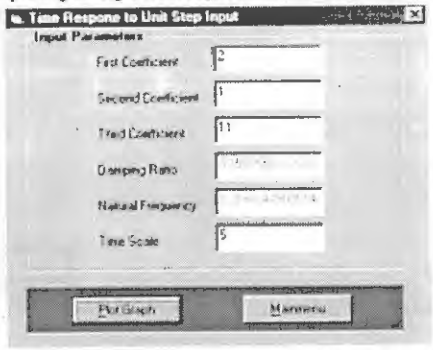


Figure 3: Input Parameter Window for Time Response Graph

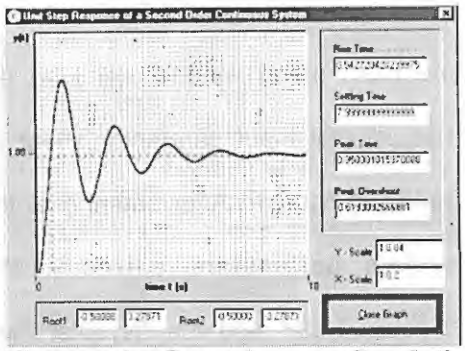


Figure 4: Sample result of the time-response module

5.2 ROOT-LOCUS MODULE

The Root Locus input parameter is loaded when the "Root Locus" button is clicked on the main menu window. Figure 5 shows the input parameter window for the root locus module. Entering the input parameters and clicking on the "Sketch Locus" button hides the input parameter window and loads the root locus graph window shown in figure 6:

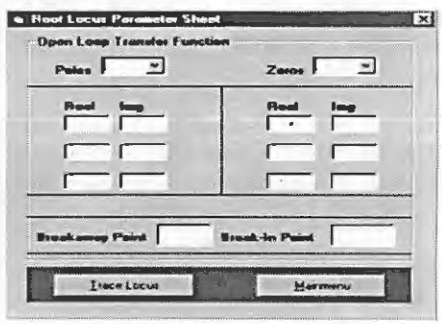
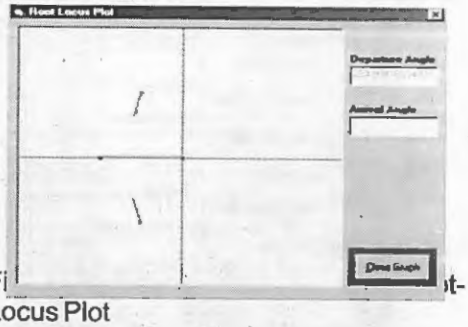


Figure 5: Input Window for Root Locus Plot

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.



Fi t-Locus Plot

5.3 NYQUIST PLOT MODULE

Clicking on the "Nyquist Plot" button on the main menu window will load the Nyquist plot input parameter sheet as shown in figure 7. After entering all the system parameters correctly, the "Nyquist Path" and the "Nyquist Plot" buttons are enabled. The user is to click on any of these buttons in order to view either the Nyquist path or the polar plot of the system. Figure 8 shows the Nyquist Path window and figures 9 11 show what the Nyquist plot looks like.

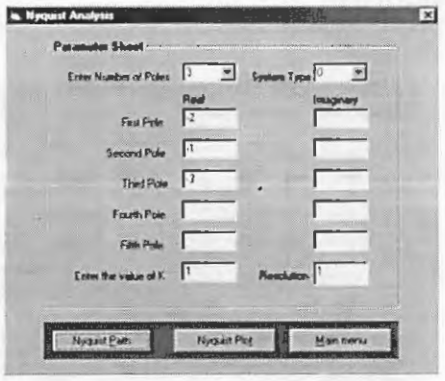


Figure 7: Open-Loop Transfer Function Input Window

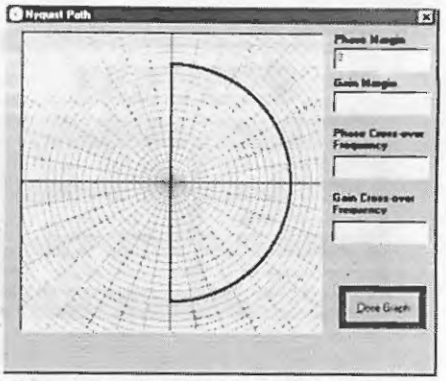


Figure 8: Nyquist Path Window

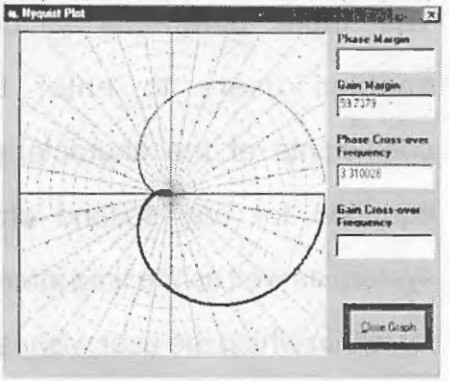


Figure 9: Sample result of the Nyquist Plot of a Type-0 system.

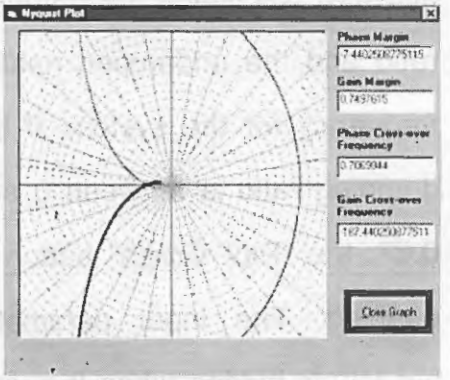


Figure 10: Nyquist Plot of a Type-1 system

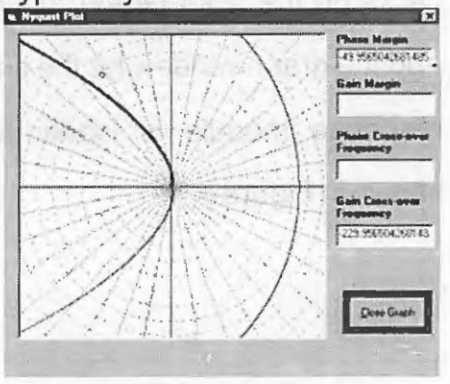
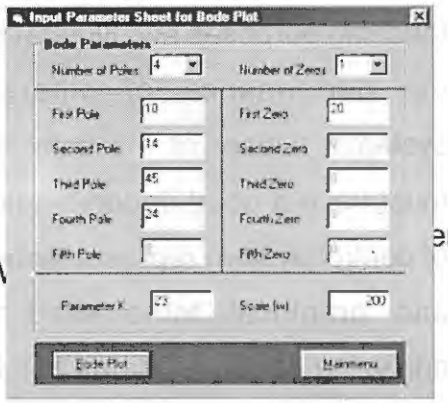


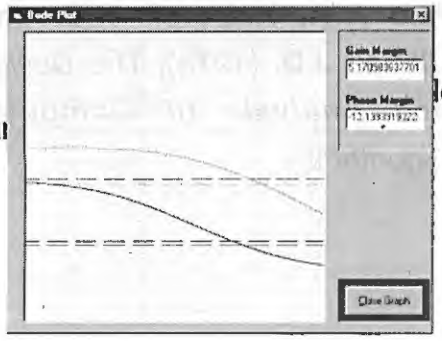
Figure 11: Nyquist Plot of a Type-2 system

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

5.4 BODE PLOT MODULE

Just like the other modules already mentioned, the Bode plot module also has an input parameter window and a graph window as displayed below in figures 12 and 13. The "Bode Plot" button is enabled once all the input parameters have been correctly entered. Clicking this button hides the input parameter window and loads the bode plot graph window for the bode magnitude plot and phase plot to be viewed.





5.5 NICHOLS CHART MODULE

The "Nichols Plot" button on the window takes the user to the Nichols input parameter window as shown in figure 14. The user is required to enter the number of poles and zeros in the systems open loop transfer function as well as the values of these parameters. The Value for the system gain K is also entered after which the user clicks on the "Nichols Plot" button, which would have been enabled immediately all the parameters have been correctly entered. On clicking this button, the input parameter sheet is hidden and the Nichols Plot graph window is loaded to view the plot.

The graph window shows the plot of the system whose parameters had been entered in the previous interface. This window also has two input fields that allow the user to plot different M and N circles, which form the chart for different values of M and N as entered by the user. This is illustrated in figure 15. Individual M and N circles are plotted each time the "Trace Chart" button is clicked with a new value of M or N.

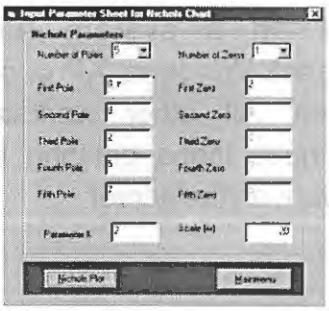


Figure 14: Input Parameter Sheet for Nichols Plot

Figure 14: Input Parameter Sheet for Nichols Plot

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

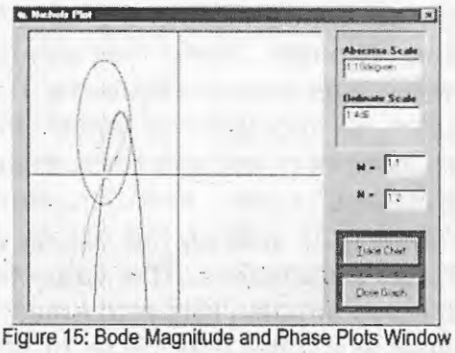


Figure 15: Bode Magnitude and Phase Plots Window

6.0 SYSTEM DEPLOYMENT

After the successful completion of the design implementation stage, the software was compiled into an executable file and deployed on the computer network server in the Department of Electrical Engineering, University of Ilorin, from where it can be accessed from any workstation within the department local area network. The software is presently undergoing a period of criticism from user, which will enable it to be upgraded based on some facilities which users expect to see on it that are not currently present in the system. The deployment was done on a system running Microsoft Windows XP operating system. All other workstations only have icons pointing to the software on the server.

7.0 CONCLUSION

Having successfully developed software for the purpose of analyzing continuous time control systems using tools as defined in the scope of work, it can be said that objectives of this project work have been satisfied.

However, this is not to say that the software does not have its limitations just like any other software. The software is not meant for use by starters in the field of control engineering; it is expected to be used by experts for teaching purposes and illustrating the performance of different systems. A user of the software must have a good understanding of control system representations and graphical tools used in analyzing control systems before the software can be of any use.

REFERENCES

Aho A.V., Hopcroft H.E. and Ullman J.D. (1974): The *Design and Analysis of Computer Algorithms*;

ISSN: 0189-9546
Faculty of Engineering & Technology University of Ilorin, Ilorin, Nigeria.

Addison-Wesley Publishing Company, Essex, England.

Aptech Limited (1999): *Programming Skills and the Internet (First Edition)*, Aptech, India.

Brassard G. and Bratley P. (1996): *Fundamentals of Algorithms*; Prentice Hall, New Jersey, USA.

Nwadiani M. and Apotu N.E. (2002): *Academic Staff Turnover in Nigerian Universities*; Education, Vol. 123.

Oni B. (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.