LAUTECH JOURNAL OF ENGINEERING AND TECHNOLOGY 2(2) 2005: 16 – 22

APPLICATION OF NUMERICAL SOFTWARES IN ELECTRICAL MACHINES MODELLING

O. I. Okoro

Department of Electrical Engineering, University of Nigeria, Nsukka, Enugu

ABSTRACT

Numerical simulation softwares are used both for research and teaching, to allow a good comprehension of the systems under study before practical implementations. In electrical machines modelling, the resulting system of differential equations are usually non-linear. Attempt to solve the equations analytically is usually difficult; thus numerical method of analysis become highly imperative. This paper illustrates the way some commercially available software packages, such as MATLAB, SIMPLORER and RMxprt are used to simulate models and their ability to make convoluted analysis rather very simple. Some practical examples are used to illustrate their capabilities in area of electrical machines and power electronics.

Keywords: Modelling, Electrical Machines, Numerical Softwares, RLC Network, State-Variable

INTRODUCTION

The gross under funding of Nigerian Universities has adversely affected research efforts in our nation's engineering faculties. Research facilities are in short supply and in some cases non-existent. Where the former is the case, the amount of research in those Universities is better imagined than described. The equipments are not only obsolete but also in limited supply to the extent that the student-equipment ratio is usually alarming. In such situation, there is an urgent need to look at alternative method of research which will not only conserve the existing facilities but also reduce the enormous effort wasted in design and testing of results. System modelling and simulation of the actual physical system offer a better alternative. Simulation can be very useful in many scientific and engineering researches that proceed as follows (Ong, 1997):

- Observing the physical system
- Formulating a hypothesis or mathematical model to explain the observation
- Predicting the behaviour of the system from solutions or properties of the mathematical model.
- Testing the validity of the hypothesis or mathematical model.

Depending on the nature of the actual physical system and the purpose of the simulation, the definitions of modelling and simulation will vary. Generally, simulation is a technique that entails setting up a model of a real situation and performing experiments on the model. The procedure for developing a model often take the following forms:

- Identifying the purpose of the model as to be able to impose on it any simplifying assumptions.
- Determining the means (instrumentation requirements) of obtaining parameters for the model.
- Defining the available computational facilities.

It is important to note that simplicity is the hallmark of most good models. The results of Okoro (Okoro, 2003; Okoro, 2002) amply attest to that. Whereas oversimplification and omissions may lead to unacceptable loss of accuracy as can be seen in (Okoro, 2002) where a great error existed in the conventional model as compare to when saturation and skin effects were included in the modelling of induction machine. However, a compromise between an oversimplified model and a detailed model must be reached as the later can be cumbersome to use both in computer simulation and hardware implementation (Krause, 1965; Barrade, 2001). In this paper, the need for numerical softwares in the modelling of electrical machines is highlighted. The paper describes three well-known software packages: MATLAB, SIMPLORER and RMxprt. Some practical examples are used to validate the enormous potentials of the aforementioned softwares. Lastly, relevant conclusion is made and a way forward proposed in engineering research in the Nigerian Universities and power industries.

SIMULATION TOOLS

The modern method of system analysis and/or design is the state variable method. The method allows an nth-order, continuous system to be represented by a set of n simultaneous, first-order differential equations. The differential equations of a lumped linear network can be written in the form (Perdikaris, 1996).

$$\dot{\mathbf{x}}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \qquad (1)$$

Where,
A = nxn System matrix
B = nxmControl matrix
x(t) = nx1 State variable vector
u(t) = Pix1 Input vector
n = System order
Pi = Number of inputs
Equation (1) represents the state variable equation of the system. The merit of the state-variable method is that it can be adapted to the compact matrix notation and matrix methods, which greatly facilitate the study of complex systems and lead to mathematical formulations that are suitable for computer solution. Again, the state variable description of a system is more general and can be applied to systems with several inputs and/or outputs, as well as to certain types of nonlinear and time-varying systems. This method has therefore been used extensively to represent electrical machine connected to a source as: (Okoro, 2002)

$$\mathbf{V} = \mathbf{R}i + \mathbf{L}\dot{i} \qquad (2)$$

Equation (2) is transformed into state-variable form as in equation (3),

$$\dot{i} = \mathbf{L}^{-1}\mathbf{V} - \mathbf{L}^{-1}\mathbf{R}i \qquad (3)$$

Equation (3) represents the state equations with current as state variables. In order to calculate the variables for a given condition of operation, the state equations are used in conjunction with the control variables (Excitation or Torque) applied externally, as well as the relevant initial conditions. For a physical machine where the rotor speed changes with time, analytical solution for equation (3) is not possible. Therefore, and efficient and a suitable numerical software packages for the simulation of the system behaviours become imperative. MATLAB, SIMULINK, SIMPLORER and Maxwell, RMxprt are, to mention but a few, commercially available software packages that help to simulate both the dynamic and steady states behaviours of electrical machines. MATLAB, licensed by Mathworks [The Mathworks®, 1997) provides a powerful matrix analysis environment, the basis of state-variable modelling of dynamic systems, for systems identification, engineering graphics, modelling and algorithm development. MATLAB has an open system environment which provides access to algorithms and source code and allows the user to mix MATLAB with FORTRAN or C language, and generates code to be used in an existing program. MATLAB provides a wide range of numerical sources with over 200 reliable, accurate and efficient mathematical subprograms which provide solutions to a broad range of mathematical problems such as matrix algebra, complex arithmetic, eigenvalues, eigenvectors, linear and non-linear systems, differential equations, and many special functions. By so doing, the time and energy that could have been used in software development are drastically reduced. The extensibility of MATLAB can be seen through the use of m-files to implement a particular function that is not built into the interpreter. Many of the functions that are supplied with the basic MATLAB release are implemented in m-files and stored in a MATLAB toolbox directory. An input template for a motor-design program called RMxprt, licensed by Ansoft corporation (The Ansoft®, 1999), helps motor designers find geometry and material characteristics that meet a design specification. The software is based on classical equations and the developers experience. In applying this program in machine design, a designer has to select a motor of interest and progress to more specific inputs from templates that include characteristics such as lamination geometry, winding configuration, materials, and a mechanical load. Modern computers crunch through these calculations in seconds. Performance plots and parameters are output for steady state, non-load magnetic, full load, locked rotor, breakdown, and rated operation. Once output plots satisfy initial design requirements, automatically generated geometry can be imported to several Finite Element Analysis, FEA tools. Consequently, the program becomes a pre-processor for FEA.
SIMPLORER is specifically developed for power electronics and drive system simulation. Several features support an easy and fast model generation and provide exceptional simulation speed even for large-scale problems. SIMPLORER is based on a unique simulator coupling technology (SIMEC®, 1999: Circuit simulator, Block diagram simulator and state machine simulator. Figure 1 outlines the steps employed by SIMPLORER to solve a problem.

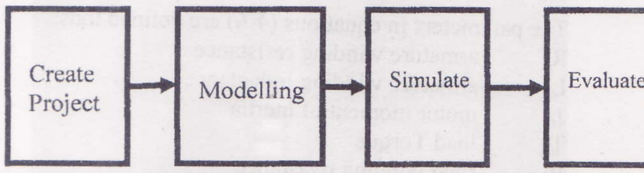

Figure 1: Solving problem in SIMPLORER.

- Create Project:

Create and manage project-files with the SIMPLORER-Simulation Center (SSC) **Commander.**

- Modelling:

Model the task using the **SIMPLORER Text Editor** or the graphical input (Schematic).

- Simulate:

Calculate the created simulation model using the simulator and display results with the **View Tool.**

- Evaluate:

Evaluate and analyze the simulation data in the **Postprocessor DAY** and/or exchange data with external programs (e.g. RMxprt, SIMULINK, Maxwell, etc).

ILLUSTRATIVE SIMULATION RESULTS

Figure 2 shows the circuit diagram of a separately excited dc motor. The electrical and mechanical model of the motor are given in Krause (Krause, 1986) as:

$$\begin{bmatrix} \dot{i}_a \\ \dot{i}_f \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-K_b\omega_m}{L_a} \\ 0 & \frac{-R_f}{L_f} \end{bmatrix} \begin{bmatrix} i_a \\ i_f \end{bmatrix} + \begin{bmatrix} \frac{V_t}{L_a} \\ \frac{V_f}{L_f} \end{bmatrix} \quad (4)$$

$$T_e = K_b i_a i_f \quad (5)$$

$$\dot{\omega}_m = \frac{1}{J_m}(T_e - B_o\omega_m - T_L) \quad (6)$$

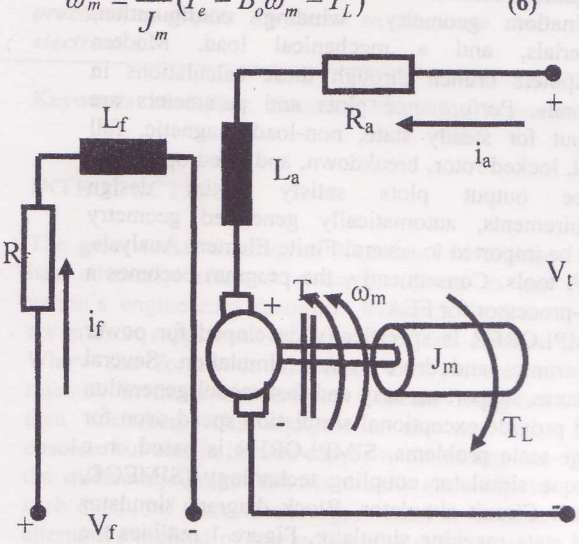

Figure 2: A separately excited DC motor.

The parameters in equations (4-6) are defined thus:

- R_a armature winding resistance
- L_a armature winding inductance
- J_m motor moment of inertia
- T_L load Torque
- R_f field winding resistance
- L_f field winding inductance
- V_f field voltage
- V_t armature voltage
- T_e electromagnetic Torque
- B_o viscous friction coefficient
- K_b induced emf constant
- ω_m mechanical rotor speed

In order to study the dynamics of the motor, the motor parameters of Table 1 is used (The Ansoft, 1997).

Table 1: Motor Parameters.

R_a	0.013Ω
L_a	0.01H
R_f	1.43Ω
L_f	0.1670H
J_m	0.210Kg-m^2
B_o	1.074e-6Nms^2
T_L	2.493Nm
K_b	0.004N-m/A^2
V_f	12V
V_t	24V

The motor startup characteristics are to be studied with V_f and V_t applied at the terminals at t=0. MATLAB programs are written as shown in Table 2. The program accepts as its input the motor parameters of Table 1 and upon simulation gives the armature current, the field current, rotor speed and the electromagnetic torque of the motor as shown in Figure 3.

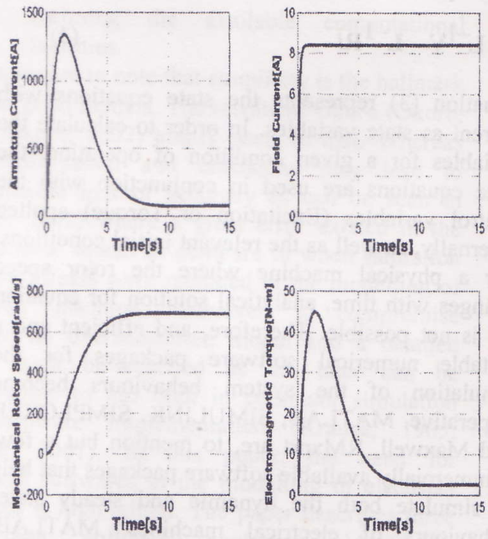

Figure 3: Startup characteristics of DC motor.

Table 2: MATLAB program for the DC motor.

```
%***************************************
***************************************
% This program is a function program. The name is
%modelNse1.m                                    *
%It solves the DC Motor differential equations
%using the                                      *
%State variable equation Xdot=AX +BU
%The program is simulated with %modelNse2.m

%***************************************
***************************************
function Idot=modelNse1(t,y)
global to tf tspan yo TL Lf Ra Rf Vt Vf Bo La Te
Kb Jm
Idot=zeros(3,1);
Idot(1)=Vt/La-Ra*y(1)/La-Kb*y(3)*y(2)/La;
Idot(2)=-Rf*y(2)/Lf+Vf/Lf;
Te=Kb*y(1)*y(2);
Idot(3)=1/Jm*(Te-Bo*y(3)-TL);

%***************************************
***************************************
% This program is a calling program. The name is
%modelNse2.m                                    *
%It calls the function program of modelNse1
                                                *
%and plots the resulting solutions of the State
%variables                                      *
%***************************************
***************************************

global to tf tspan yo TL Lf Ra Rf Vt Vf Bo La Jm
Te Kb
%INITIAL STATES VECTOR
yo=[0 0 0];
%MOTOR PARAMETERS
TL=2.493;
Bo=1.074e-6;
Vf=12;
Vt=24;
Kb=0.004;
Jm=0.210;
Lf=0.167;
Rf=1.43;
La=0.01;
Ra=0.013;
%INITIAL AND FINAL TIME
to=0.0;
tf=15;
tspan=to:0.001:tf;
%CALLING THE FUNCTION PROGRAM
modelNse1.
[t,y]=ode45('modelNse1',tspan,yo);
%COMPUTE ELECTROMAGNETIC TORQUE.
Te=Kb*y(:,1).*y(:,2);
%PLOTTING GRAPHS USING SUBPLOT.
subplot(2,2,1)
plot(t,y(:,1),'r')
grid on
xlabel('Time[s]')
ylabel('Armature Current[A]')
subplot(2,2,2)
plot(t,y(:,2),'b')
grid on
xlabel('Time[s]')
ylabel('Field Current[A]')
subplot(2,2,3)
plot(t,y(:,3),'g')
grid on
xlabel('Time[s]')
ylabel('Mechanical Rotor Speed[rad/s]')
subplot(2,2,4)
plot(t,Te,'r')
grid on
xlabel('Time[s]')
ylabel('Electromagnetic Torque[N-m]')
```

The second simulation example illustrates the use of RMxprt to design a 3-phase, 50Hz, 4-Pole, 4.8KW squirrel-cage induction machine. RMxprt uses a combination of analytical and magnetic circuit equations to predict the performance of electrical motor. In carrying out the design, the first step is to select the type of motor to be designed(In this case Three Phase Induction Motor). A motor specific template appears where specific stator, rotor, material, and winding parameters as shown in Figure 4 are entered.

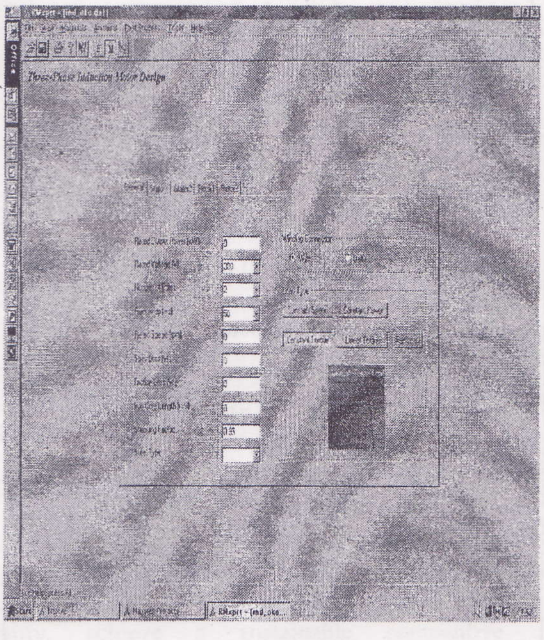

Figure 4: Template to enter desired stator, rotor, material and winding parameters.

These parameters completely characterize the design and the simulation which will be completed. The motor specific template includes a standard material database for easy selection of materials.

The user may also enter B-H data for materials not included in the database. Once the template is complete, RMxprt runs the analysis. Within seconds, RMxprt generates detailed performance data in the form of a design sheet detailing input parameters and relevant output parameters as shown in Table 3.

Table 3: Generated performance data.

Three-Phase Induction Motor Design	
File: c:/ansoft/ind_oko.pjt/ind_oko.res	
GENERAL DATA	
Given Output Power (kW):	4.8
Rated Voltage (V):	380
Winding Connection:	Delta
Number of Poles:	2
Given Speed (rpm):	1450
Frequency (Hz):	50
Stray Loss (W):	100
Friction and Wind Loss (W):	48
Type of Load:	Constant Torque
Iron Core Length (mm):	175
Stacking Factor of Iron Core:	0.93
Type of Steel:	D23
STATOR DATA	
Number of Stator Slots:	36
Outer Diameter of Stator (mm):	200
Inner Diameter of Stator (mm):	125
Type of Stator Slot:	3
Dimension of Stator Slot	
hs0 (mm):	0.6
hs1 (mm):	2.71
hs2 (mm):	14.19
bs0 (mm):	2.5
bs1 (mm):	5.42
bs2 (mm):	7.76
rs (mm):	1.2
Number of Conductors per Slot:	36
Number of Parallel Branches:	2
Number of Wires per Conductor:	1
Type of Coils:	21
Coil Pitch:	18
Wire Diameter (mm):	0.71
Wire Wrap Thickness (mm):	0.07
Slot Insulation Thickness (mm):	0.35
Top Free Space in Slot (%):	1
Bottom Free Space in Slot (%):	1
Conductor Length Adjustment (mm):	0
ROTOR DATA	
Number of Rotor Slots:	30
Air Gap (mm):	0.45
Inner Diameter of Rotor (mm):	28
Type of Rotor Slot:	4
Dimension of Rotor Slot	
hr0 (mm):	2
hr01 (mm):	0
hr2 (mm):	13.5
br0 (mm):	1
br2 (mm):	4.5
rr (mm):	0
Skew Width (mm):	10.796
Height of End Ring (mm):	13.5
Width of End Ring (mm):	4.5
Resistivity of Rotor Conductor at 75 C (ohm.mm^2/m):	0.0217
RATED-LOAD OPERATION	
Stator Resistance (ohm):	3.04848
Stator Leakage Reactance (ohm):	1.09126
Rotor Resistance (ohm):	1.11452
Rotor Leakage Reactance (ohm):	1.2753
Resistance Corresponding to Iron-Core Loss (ohm):	664.406
Magnetizing Reactance (ohm):	43.4193
Stator Phase Current (A):	13.1749
Current Corresponding to Iron-Core Loss (A):	0.510744
Magnetizing Current (A):	7.81544
Rotor Phase Current (A):	9.8252
Copper Loss of Stator Winding (W):	1587.44
Copper Loss of Rotor Winding (W):	322.77
Iron-Core Loss (W):	519.95
Friction & Wind Loss (W):	48
Stray Loss (W):	100
Input Power (kW):	12.2029
Output Power (kW):	9.6247
Mechanical Shaft Torque (N.m):	31.6638
Efficiency (%):	78.8725
Power Factor:	0.805818
Rated Slip:	0.0322917
Rated Shaft Speed (rpm):	2903.13
NO-LOAD OPERATION	
Stator Resistance (ohm):	3.04848
Stator Leakage Reactance (ohm):	1.09216
Rotor Resistance (ohm):	1.11447
Rotor Leakage Reactance (ohm):	1.29088
Phase Current (A):	8.49851
Iron-Core Loss (W):	611.653
Input Power (W):	1432.65
Power Factor:	0.137553
Slip:	0.000165842
Shaft Speed (rpm):	2999.5

RMxprt also generates graphical waveforms such as current, torque, efficiency, etc as shown in Figure 5.

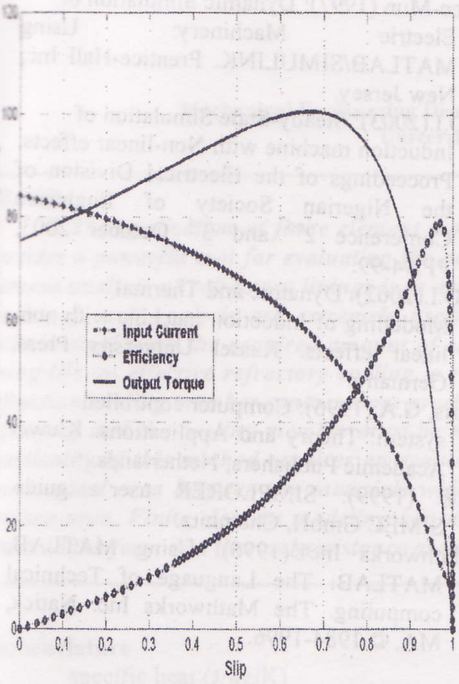

Figure 5 : Steady-state performance curves of Induction machine.

The versatile plot data function allows any parameter to be plotted against any other parameter. The third application presents the modelling of a series R-L-C network using SIMPLORER. The model equation, representing the transient behaviour of the RLC network is given as (Nilsson, 1986):

$$Ri + L\frac{di}{dt} + V_C = V_S \qquad (7)$$

$$i = C\frac{dV_C}{dt} \qquad (8)$$

In state variable form, equations(7-8) become:

$$\begin{bmatrix} \dot{V_C} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ \frac{-1}{L} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} V_c \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} [V_s] \qquad (9)$$

Figure 6 shows the SIMPLORER model for the RLC network carried out using the INTERN2 Library in SIMPLORER Schematic.

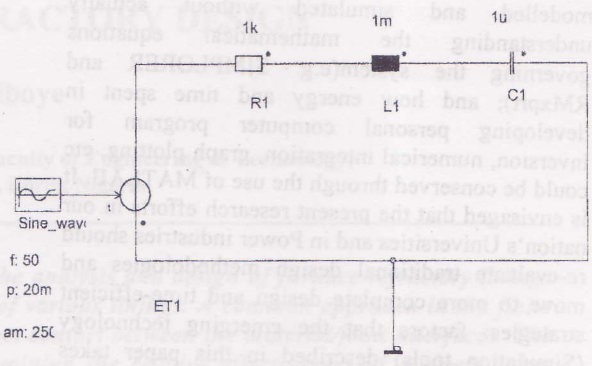

Figure 6: SIMPLORER Model of RLC Network.

After defining the outputs for the simulation, the simulation is started and the results which show the source voltage, the capacitor voltage, the resistor voltage and the inductor voltage are presented in Figure 7.

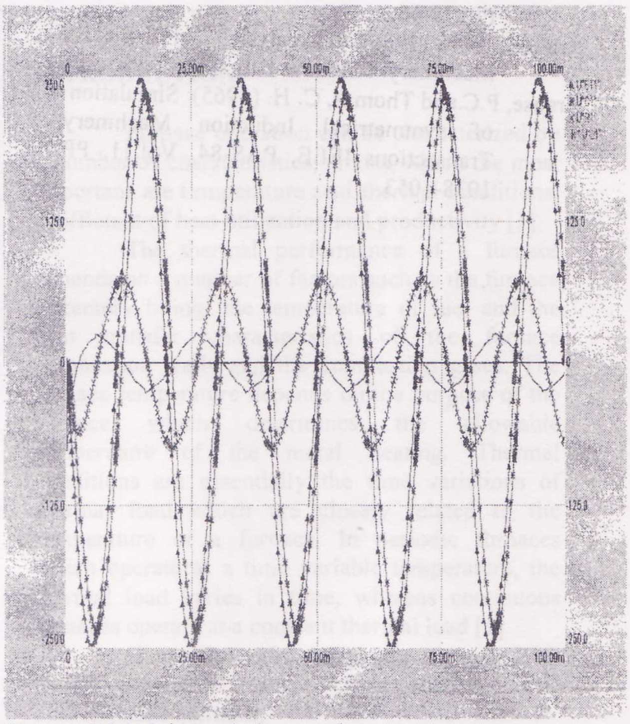

Figure 7: Model simulation results.

CONCLUSION

Electric machine manufacturers and industrial engineers that find ways to minimize engineering time, reduce prototyping cost and optimise product quality, achieve a sustainable competitive advantage. The best way to actualise these objectives is to use simulation software to predict electric machine performance without creating prototypes. The practical applications presented in this paper show such advantage and clearly elucidates how a complicated system can be modelled and simulated without actually understanding the mathematical equations governing the system(e.g. SIMPLORER and RMxprt); and how energy and time spent in developing personal computer program for inversion, numerical integration, graph plotting, etc could be conserved through the use of MATLAB. It is envisaged that the present research efforts in our nation's Universities and in Power industries should re-evaluate traditional design methodologies and move to more complete design and time-efficient strategies: factors that the emerging technology (Simulation tools) described in this paper takes credit.

REFERENCES

Ansoft® (1999): The Maxwell user's guilde. Ansoft corporation, Pittsburgh, © 1998-1999.

Barrade, P. (2001): Simulation tools for power electronics: Teaching and Research. SIMPLORER Workshop 2 001, C hemnitz, PP..35-46.

Krause, P.C.(1986): Analysis of Electric Machinery. McGraw-Hill, New York.

Krause, P.C.and Thomas, C. H. (1965): Simulation of symmetrical Induction Machinery. Transactions IEEE, PAS-84, Vol.11, PP. 1038-1053.

Nilsson, J. W.(1986): Electric Circuits. Addison-Wesley publishing company, England.

Ong, Chee-Mun (1997): Dynamic Simulation of Electric Machinery: Using MATLAB/SIMULINK. Prentice-Hall Inc; New Jersey.

Okoro, O.I.(2003): Steady-State Simulation of Induction machine with Non-linear effects. Proceedings of the Electrical Division of the Nigerian Society of Engineers Conference 2nd and 3rd October 2003, PP.94-99.

Okoro, O.I.(2002): Dynamic and Thermal Modelling of Induction machine with non-linear effects. Kassel University Press, Germany.

Perdikaris, G.A.(1996): Computer controlled system: Theory and Applications. Kluwer Academic Publishers, Netherlands.

SIMEC® (1999): SIMPLORER user's guide. SIMEC GmbH, Chemnitz.

The Mathworks Inc®(1996): Using MATLAB, MATLAB: The Language of Technical computing. The Mathworks Inc. Natick, MA, © 1984-1996.