

DEVELOPMENT OF THE COMPUTER-AIDED DIAGRAM (CAD) FOR
ELECTRICAL MACHINE FOR UNDERGRADUATE PURPOSE
LEARNING

MOHD SYAFIQ MOHD FUAD

UNIVERSITY MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: DEVELOPMENT OF THE COMPUTER-AIDED DIAGRAM (CAD) FOR ELECTRICAL MACHINE FOR UNDERGRADUATE PURPOSE LEARNING

SESI PENGAJIAN: 2007/2008

Saya MOHD SYAFIQ MOHD FUAD (850911-14-5683)
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan & Teknologi Malaysia.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

NO. 47, JALAN BATU MUDA 7,
TAMAN BATU MUDA, BATU CAVES,
68100 KUALA LUMPUR,
W. P. KUALA LUMPUR.

MISS NAJIDAH BINTI HAMBALI
(Nama Penyelia)

Tarikh: _____

Tarikh: : _____

- CATATAN:
- * Potong yang tidak berkenaan.
 - ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
 - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“I hereby acknowledge that the scope and quality of this thesis is qualified for the
award of the Bachelor Degree of Electrical Engineering (Electronic)”

Signature : _____

Name : MISS NAJIDAH BINTI HAMBALI

Date : _____

DEVELOPMENT OF THE COMPUTER-AIDED DIAGRAM (CAD) FOR
ELECTRICAL MACHINE FOR UNDERGRADUATE PURPOSE LEARNING

MOHD SYAFIQ MOHD FUAD

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Electronic)

Faculty of Electrical & Electronics Engineering
University Malaysia Pahang

MAY 2008

“All the trademark and copyright use herein are property of their respective owner. Reference of information from other sources is quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : MOHD SYAFIQ MOHD FUAD

Date : _____

*Specially dedicated to
My beloved parent*

ACKNOWLEDGEMENT

First and foremost, I am very grateful to the almighty ALLAH S.W.T for giving me this opportunity to accomplish my Final Year Project.

In the preceding of this project and thesis, I have established a contact with many people who have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my project supervisor, Ms. Najidah Binti Hambali, for encouragement, guidance, motivation and friendship. Without her unlimited support and interest, this project and thesis would never be completed

My fellow friends and colleagues should also be recognized for their support. My sincere appreciation also extends to all lecturer and others who had provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space.

Most importantly I extend my gratitude to my parents who have encouraged me throughout my education and I will always be grateful for their sacrifice, generosity and love.

ABSTRACT

This study developed a system analyses the single phase transformer by generates the equivalent circuit and solve the characteristic such as efficiency of the transformer. This project will be used in undergraduate purpose learning and it will be carried out by using graphical user interface (GUI) of Visual Basic (VB) programming language environment. This system can be divided into two sections which are to obtain the equivalent circuit of the transformer and to obtain the efficiency of the transformer. First stage of the program will prompt user to key in the results from transformer test which are the open-circuit test (no-load test) and short circuit test. Program will analyze the data and the equivalent circuit of the transformer will be display as a results. This function returns the referred equivalent circuit for primary and secondary winding. After equivalent circuits are obtained, in second stage of the program, the user is prompted to enter the load specification to obtain the transformer performance characteristic including an efficiency curve from 25 to 125 percent of full-load.

ABSTRAK

Kajian ini memajukan satu sistem menganalisa pengubah satu fasa dengan menghasilkan litar setaraf dan menyelesaikan sifat seperti kecekapan pengubah tersebut. Projek ini akan digunakan dalam tujuan pembelajaran palajar prasiswazah dan akan dihasilkan dengan menggunakan antaramuka grafik pengguna iaitu Visual Basic (VB) persekitaran bahasa pengaturcaraan. Sistem ini boleh dibahagikan kepada dua bahagian iaitu untuk memperoleh litar setaraf pengubah dan memperoleh kecekapan pengubah. Peringkat pertama program ini akan meminta pengguna memasukkan keputusan daripada ujian pengubah iaitu ujian litar terbuka dan ujian litar tertutup. Program akan menganalisa data dan litar setara pengubah akan dipamerkan sebagai hasil analisa. Peranan ini menghasilkan litar setara pengubah merujuk kepada belitan utama dan sekunder. Selepas memperoleh litar setara, dalam fasa kedua program tersebut, pengguna diminta memasukkan spesifikasi beban untuk mendapatkan sifat prestasi pengubah termasuk garis lengkungan kecekapan dari 25 hingga 125 peratus dari beban penuh.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF FIGURES	x
	LIST OF ABREVIATION	xii
	LIST OF APPENDICES	xiii
1	INTRODUCTION	1
	1.1 Overview	1
	1.2 Objectives	2
	1.3 Scopes	2
	1.3.1 Transformer Analysis	2
	1.3.2 Develop the GUI Using Visual Basic	3
2	LITERATURE REVIEW	4
	2.1 Hadi Saadat Software	4
	2.2 Transformer	6
	2.2.1 Transformer Operation	6
	2.2.2 Voltage, Current and Impedance	7

	Transformations	
	2.2.3 Nonideal Transformer	8
	2.2.4 Equivalent Circuits	8
	2.2.5 Tests on Transformers	10
	2.2.5.1 Open-Circuit Test	10
	2.2.5.2 Short-Circuit Test	11
	2.3 Visual Basic	13
3	METHODOLOGY	15
	3.1 Work Flow Chart	15
	3.1.1 The Program	17
	3.1.1.1 Building Visual Basic Program	17
	3.1.1.2 Compile the Program	17
	3.1.1.3 Test the Program	18
	3.1.2 The Graphical User Interface	18
	3.1.2.1 Develop the GUI Using Visual Basic	18
	3.1.2.2 Displaying the Data in the GUI	19
	3.1.2.3 Test the GUI	19
4	RESULTS AND ANALYSIS	20
	4.1 Result of the Visual Basic Program	20
	4.1.1 Transformer Tests	22
	4.1.2 Equivalent Circuit of the Transformer	22
	4.1.3 Transformer Efficiency	23
	4.2 Overall of Project Operation	26
5	CONCLUSIONS AND RECOMMENDATION	28
	5.1 Summary of the Work	28
	5.2 Recommendations for Future the Work	28

REFERENCES	30	
APPENDICES	31	
APPENDIX A	Original Program Results	32
APPENDIX B	Customized Program Results	39

LIST OF FIGURES

FIGURE NO	TITLE	PAGE
1.1	The equivalent circuit of the transformer referred to primary	1
2.1(a)	Show the example of a MATLAB program	5
2.1(b)	Show the example of a MATLAB program	5
2.2	Basic Component of the transformer	6
2.3(a)	Ideal transformer	9
2.3(b)	Nonideal transformer	9
2.4(a)	Equivalent circuit of a nonideal transformer referred to primary	9
2.4(b)	Equivalent circuit of a nonideal transformer referred to secondary	9
2.5	Phasor diagram	10
2.6	Equivalent circuit	12
2.7	Example for Visual Basic Programming	14
3.1	Flow chart of working progress	16
4.1(a)	Show the program main page where user prompted to enter the required data	21
4.1(b)	Show the transformer efficiency and efficiency curve from 25 to 125% full load	21
4.2	Data to be analyze	22
4.3	The equivalent circuits	22
4.4	Transformer efficiency	23

4.5(a)	Show the efficiency curve from 25 to 125 percent of full-load	25
4.5(b)	Show the efficiency from 1 to 300MVA of load	25
4.6	Block diagram of system flow	26

LIST OF ABBREVIATIONS

GUI	- Graphical User Interface
VB	- Visual Basic
MATLAB	- Matrix Laboratory
SIMULATION	- Simulation and Capture
RAD	- Rapid Application Development
DAO	- Data Access Object
RDO	- Remote Data Objects
DOS	- Disk Operating System
BASIC	- Beginners' All-purpose Symbolic Instruction Code

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Program Design	32
B	Example Program for Calculation	39

CHAPTER 1

INTRODUCTION

1.1 Overview

This study developed a system that can solve the single phase transformer problems. First, the windings of transformers have resistance and the cores are not infinitely permeable. Furthermore, the flux produce by the primary is not completely captured by the secondary. And finally, the iron cores produce eddy-current and hysteresis losses, which contribute to the temperature rise of the transformer [2]. By using this simulation, the properties of a practical transformer can be described by an equivalent circuit. This system will be carried out by using Graphical User Interface (GUI) that will be developing using Visual Basic (VB).

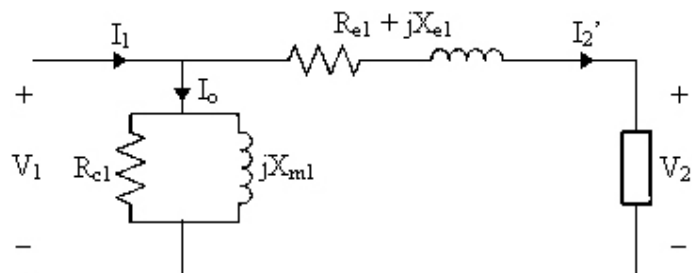


Figure 1.1: The equivalent circuit of the transformer referred to primary.

There are many reasons for using computer-aided analysis programs for analyzing circuits, such as high computational speed of digital computers that offer great saving time for large circuit analyzed. The process to obtain the transformer equivalent circuits consists of various methods of mathematical calculation which is difficult to perform by hand. The calculation can be easily done by computer which

is generated by a program developed using Visual Basic (VB). Hence errors done by manual calculation can also be eliminated. This computer-aided diagram will offer suitable medium learning for electric machine for undergraduate's purpose learning.

1.2 Objectives

There are three main objective of the project which is:-

1. To develop a system that analyses the single phase transformer and generates the equivalent circuit comprising an ideal transformer and resistance and reactance.
2. To solve the characteristic such as efficiency of the transformer.
3. To offer the suitable medium learning for electrical machine for students.

1.3 Scopes

1.3.1 Transformer Analysis

The equivalent circuit model of a transformer is shown in Figure 1.1. The equivalent circuits consist of an ideal transformer of ratio $N_1:N_2$ together with elements which represent the imperfections of the real transformer. An ideal transformer would have winding with zero resistance and a lossless, infinite permeability core [1]. In the practical transformer, there are properties must be taken into accounts which are iron losses, magnetizing reactance, leakage reactance and winding resistance.

The equivalent circuit can now be used to predict the performance characteristic of the transformer. An importance aspect is the transformer efficiency.

Another importance performance characteristic of the transformer is change in the secondary voltage from no-load to full-load. A figure to merit used to compare the relative performance of different transformer is the voltage regulation [1].

1.3.2 Develop the GUI Using Visual Basic

The development of a Graphical User Interface (GUI) is using Visual Basic. Visual Basic is the easier programming to learn. A Computer cannot understand any person's spoken language. A spoken language such as English, French, is simply too general and ambiguous for computers to understand. Therefore, we must adapt to the machine and learn the computer language so that the computer can understand [6].

CHAPTER 2

LITERATURE REVIEW

2.1 Hadi Saadat Software

Every copies of a Power System Analysis text book contain a copy of programming data to perform the fault analysis using MATLAB [1]. The programming data is to be installed in the MATLAB program to be put in the MATLAB's library.

This program is developed for obtaining the transformer characteristic. It can be used independently when the arguments of the functions are defined in the MATLAB environment. The user will be prompted to enter the required data [1]. Figure 2.1 (a) & (b) show the GUI:

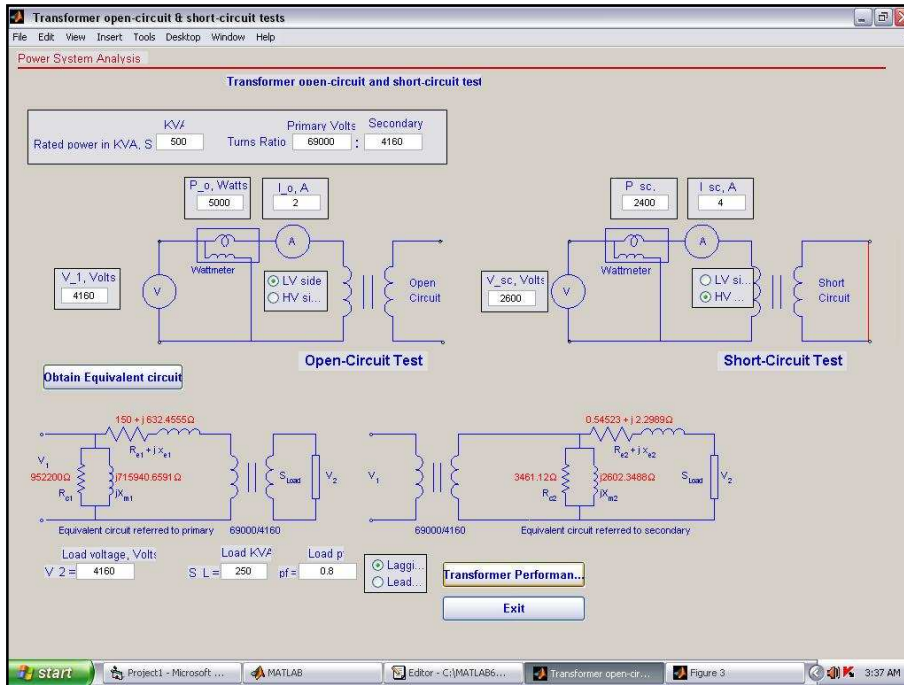


Figure 2.1 (a): Show the example of a MATLAB program.

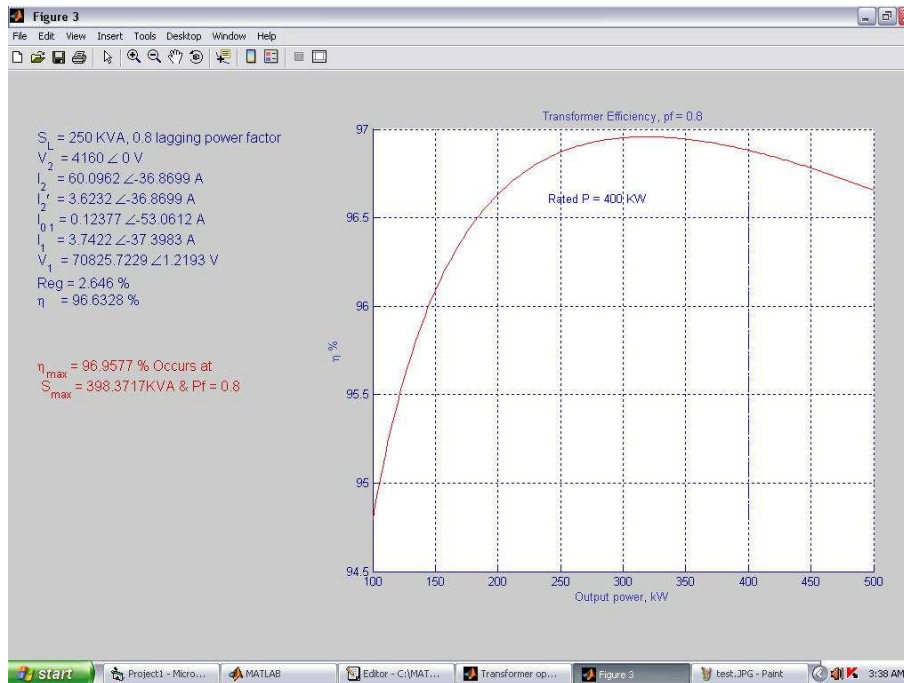


Figure 2.1 (b): Show the example of a MATLAB program.

2.2 Transformer

2.2.1 Transformer Operation

A transformer is an electromagnetic device having two or more mutually coupled windings (share a common magnetic circuit). Figure below show a two-winding, ideal transformer; the transformer is ideal in the sense that its core is lossless and is infinitely permeable, it has no leakage fluxes, and the windings have no losses [3].

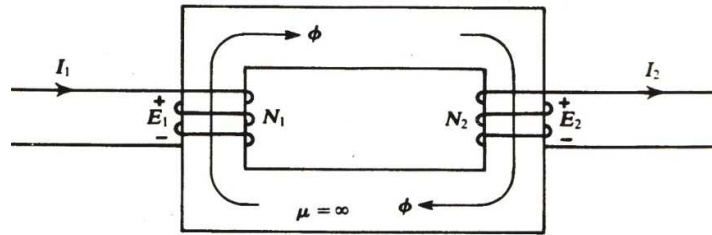


Figure 2.2

In Figure 2.2, the basic components are the core, the primary winding, N_1 and the secondary winding, N_2 . If ϕ is the mutual (or core) flux threading each turn of N_1 and N_2 , then according to Faraday's Law of electromagnetic induction, emfs e_1 and e_2 are induced in N_1 and N_2 , given by [3]

$$e_1 = N_1 \frac{d\phi}{dt} \qquad e_2 = N_2 \frac{d\phi}{dt} \qquad (4.01)$$

The direction of e is such as to produce a current that gives rise to a flux which opposes the flux change $d\phi$ (Lenz's Law). From (4.1), $e_1 / e_2 = N_1 / N_2$, or in terms of rms values,

$$\frac{E_1}{E_2} = \frac{N_1}{N_2} \equiv a \qquad (4.02)$$

where a is the turn ratio [3].

The transformer being ideal, $e_1 = v_1$ and $e_2 = v_2$ in Figure 4.2; hence the flux and voltage are related by [3]

$$\phi = \frac{1}{N_1} \int v_1 dt = \frac{1}{N_2} \int v_2 dt \quad (4.03)$$

If the flux varies sinusoidally, $\phi = \phi_m \sin \omega t$, then the corresponding induced voltage, e , in an N -turn winding is given by

$$e = \omega N \phi_m \cos \omega t \quad (4.04)$$

and the rms value of this induced voltage is

$$E = \frac{\omega N \phi_m}{\sqrt{2}} = 4.44 f N \omega_m \quad (4.05)$$

which is known as the emf equation. In (4.5), $f = \omega/2\pi$ is the flux frequency in Hz [3].

2.2.2 Voltage, Current and Impedance Transformations

Transformers are used to effect voltage, current and impedance transformations, and to provide isolation (that is to eliminate direct connections between electrical circuits). The voltage transformation property of an ideal transformer was developed in Section 4.21:

$$\frac{V_1}{V_2} = \frac{E_1}{E_2} = a \quad (4.06)$$

where the subscripts 1 and 2 respectively correspond to the primary and the secondary sides [3].

For an ideal transformer, the net mmf around its magnetic circuit must be zero. Thus, $N_1 I_1 - N_2 I_2 = 0$, or [3]

$$\frac{I_2}{I_1} = \frac{N_1}{N_2} \equiv a \quad (4.07)$$

From (4.06) and (4.07), it can be shown that if impedance Z_2 is connected to the secondary, the impedance Z_1 seen at the primary satisfies [3]

$$\frac{Z_1}{Z_2} = \left(\frac{N_1}{N_2} \right)^2 \equiv a^2 \quad (4.08)$$

2.2.3 Nonideal Transformer

An actual transformer differs from an ideal transformer in that the former has hysteresis and eddy-current (core) losses and has resistive (I^2R) losses in its primary and secondary windings. Furthermore, the core of a nonideal transformer is not perfectly permeable and requires a finite mmf for its magnetization. Also, because of leakage, not all fluxes link with the primary simultaneously in a nonideal transformer [3].

2.2.4 Equivalent Circuits

An equivalent circuit of an ideal transformer is shown in Figure 2.3(a). When the effects of winding resistances, leakage reactance, magnetizing reactance and core losses are included, the circuit of Figure 2.3(b) results, in which the primary and the secondary are coupled by an ideal transformer. The ideal transformer may be removed from Figure 2.3(b) and the entire equivalent circuit may be referred either to the primary as in Figure 2.4(a) or to the secondary as in Figure 2.4(b) [3].

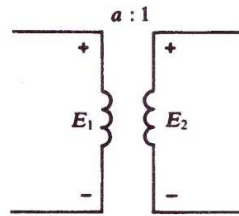


Figure 2.3 (a): Ideal transformer.

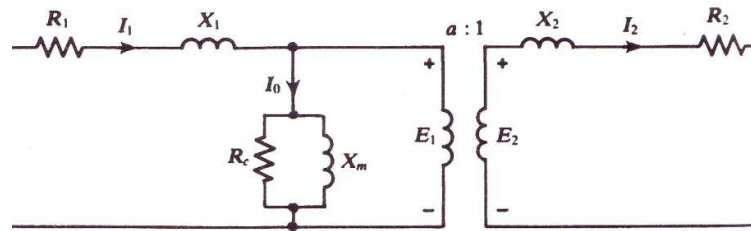


Figure 2.3 (b): Nonideal transformer.

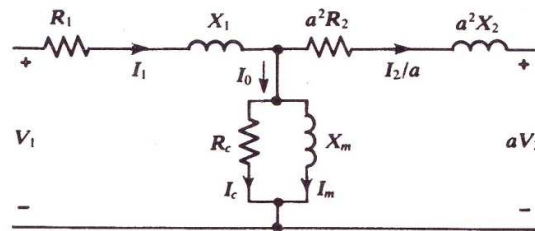


Figure 2.4 (a): Equivalent circuit of a nonideal transformer referred to primary.

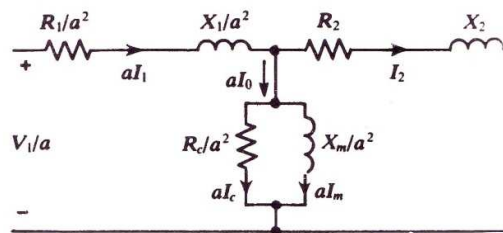


Figure 2.4(b): Equivalent circuit of a nonideal transformer referred to secondary.

A phasor diagram for the circuit of Figure 2.5, in the case of lagging power factor, is shown in figure below [3]. The various symbols in Figure 2.3, Figure 2.4 and Figure 2.5 are:

- a = turns ratio
- E_1 = primary induced voltage
- E_2 = secondary induced voltage
- V_1 = primary terminal voltage

V_2 = secondary terminal voltage
 I_1 = primary current
 I_2 = secondary current
 I_0 = no-load current
 R_1 = resistance of the primary winding
 R_2 = resistance of the secondary winding
 X_1 = primary leakage reactance
 X_2 = secondary leakage reactance
 I_m and X_m = magnetizing current and reactance
 I_c and R_c = current and resistance accounting for the core losses

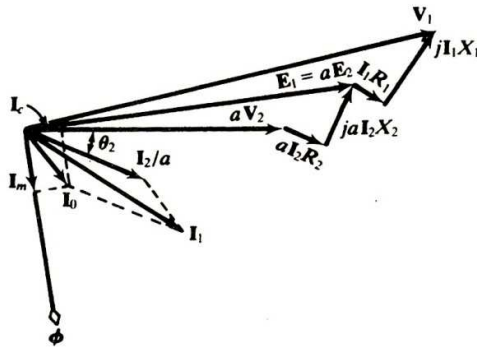


Figure 2.5

2.2.5 Tests on Transformers

Transformer performance characteristics can be obtained from the equivalent circuits of Section 4.24. The circuit parameters are determined either from design data or from test data. The two common tests are as follows [3].

2.2.5.1 Open-Circuit Test

Here one winding is open-circuited and a voltage - usually the rated voltage at rated frequency - is applied to the other winding. Voltage, current, and power at the terminals of this winding are measured. The open-circuit voltage of the second winding is also measured, and from this measurement a check on the turns ratio can be obtained. It is usually convenient to apply the test voltage to the winding that has

a voltage rating equal to that of the available power source. For step-up voltage transformers, this means that the open-circuit voltage of the second winding will be higher than the applied voltage, sometimes much higher. Care must be exercised in guarding the terminals of this winding to insure the safety of test personnel and to prevent these terminals from getting close to other electrical circuits, instrumentation, grounds and so forth [3].

In presenting the no-load parameters obtainable from test data, it is assumed that voltage is applied to the primary and that the secondary is open-circuited. The no-load power loss is equal to the wattmeter reading in this test; core loss is found by subtracting the ohmic loss in the primary, which is usually small and may be neglected in some cases. Thus, if P_o , I_o , and V_o are the input power, current, and voltage, then the core loss is given by

$$P_c = P_o - I_o^2 R_1 \quad (4.09)$$

The primary induced voltage is, in phasor form,

$$\mathbf{E}_1 = V_o \angle 0^\circ - (I_o \angle \theta_o)(R_1 + jX_1) \quad (4.10)$$

where $\theta \equiv$ no-load power-factor angle $= \cos^{-1}(P_o/V_o I_o)$. Other circuit quantities are found from [3]:

$$\begin{aligned} R_c &= \frac{E_1^2}{P_c} & I_c &= \frac{P_c}{E_1} & I_m &= \sqrt{I_o^2 - I_c^2} \\ X_m &= \frac{E_1}{I_m} & a &\approx \frac{V_o}{E_2} \end{aligned} \quad (4.11)$$

2.2.5.2 Short-Circuit Test

In this test, one winding is short-circuited across its terminals, and a reduced voltage is applied to the other winding. This reduced voltage is of such a magnitude as to cause a specific value of current - usually the rated current - to flow in the

short-circuited winding. Again, the choice of the winding to be short-circuited is usually determined by the measuring equipment available. Care must be taken to note which winding is short-circuited, for it will serve as the reference winding for expressing the impedance components obtained in the test. Here we shall let the secondary be short-circuited and the reduced voltage be applied to the primary.

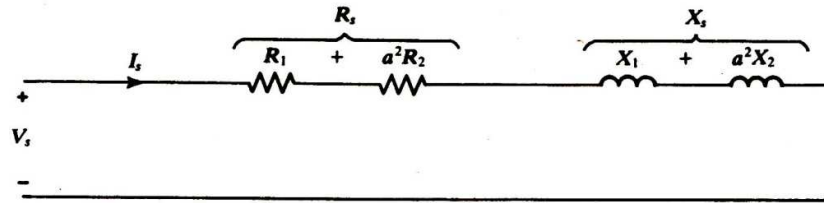


Figure 2.6

With a very low voltage applied to the primary winding, the core-loss current and magnetizing current become very small and the equivalent circuit reduces to that of Figure 2.6. Thus, if T_s , I_s and V_s are the input power, current, and voltage under short circuit, then, referred to the primary,

$$Z_s = \frac{V_s}{I_s} \quad (4.12)$$

$$R_1 + a^2 R_2 \equiv R_s = \frac{P_s}{I_s^2} \quad (4.13)$$

$$X_1 + a^2 X_2 \equiv X_s = \sqrt{Z_s^2 - R_s^2} \quad (4.14)$$

Given R_1 and $a^2 R_2$ can be found from (4.13). In (4.14) it is usually assumed that the leakage reactance is divided equally between the primary and the secondary; that is [3],

$$X_1 = a^2 X_2 = \frac{1}{2} X_s \quad (4.15)$$

2.3 Visual Basic

Visual Basic (VB) is an event driven programming language and associated development environment from Microsoft for its COM programming model. VB has been replaced by Visual Basic.NET. The older version of VB was derived heavily from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using DAO, RDO, or ADO, and creation of ActiveX controls and objects [4].

Visual Basic is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means Beginners' All-purpose Symbolic Instruction Code. It is a fairly easy programming language to learn. The codes look a bit like English Language. Different software companies produced different version of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA and so on [5].

A Computer cannot understand any person's spoken language. A spoken language such as English, French, is simply too general and ambiguous for computers to understand. Therefore, we must adapt to the machine and learn the computer language so that the computer can understand. This is where visual basic comes into it - when you type visual basic source code into the computer, the computer processes these statements into Visual Basic language. The statements in visual basic, however, do not have multiple meanings within the same context [6]. Figure 2.7 is the example for Visual Basic Programming.

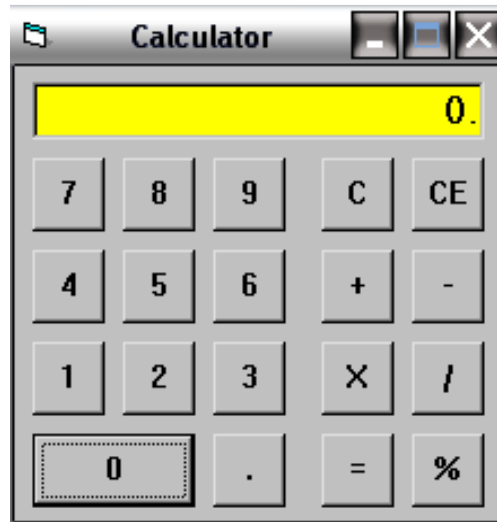


Figure 2.7

CHAPTER 3

METHODOLOGY

3.1 Work Flow Chart

This chapter will explain about design, interface and the operation of the software. Before looking at the detail of all the methods below, it is best to begin with brief review the correlation of all methods. Figure 3.1 below show the correlation of all methods in this project.

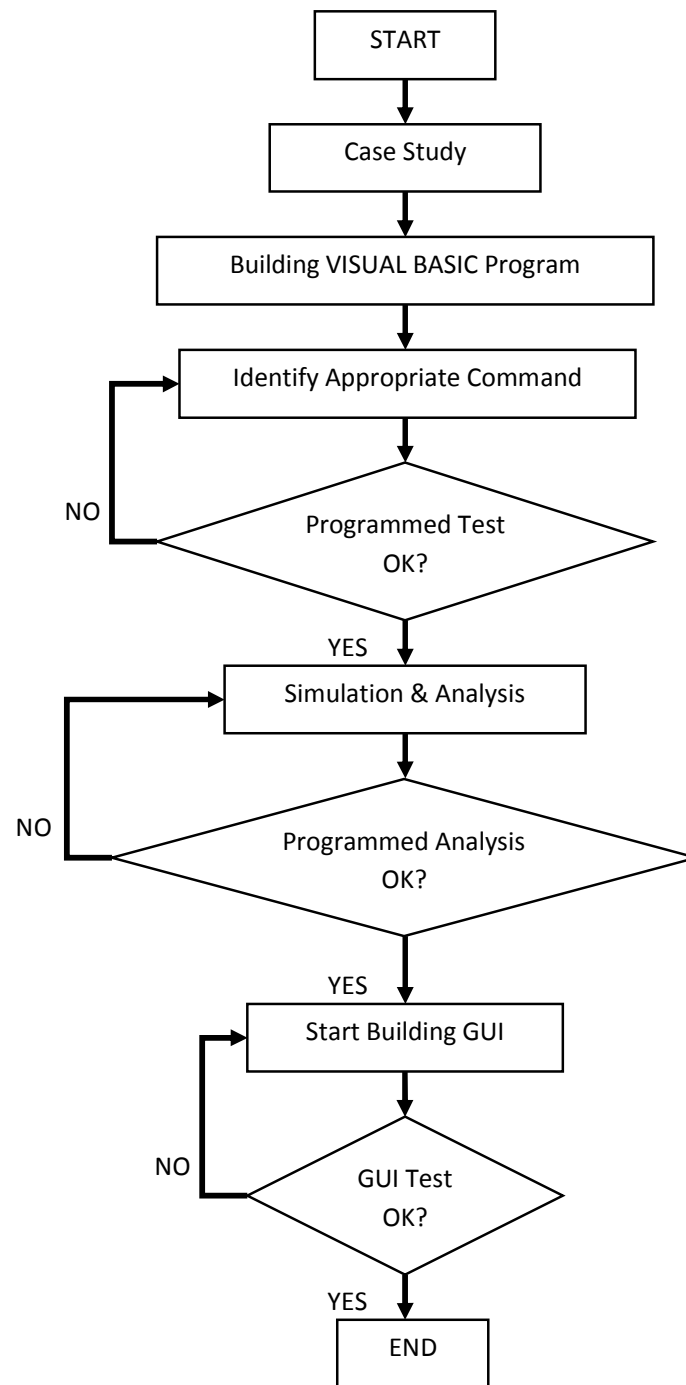


Figure 3.1: Flow chart of working progress

3.1.1 The Program

There are two parts which are the program that analyze the single phase transformer and the GUI of the program. The first task is to develop the program.

3.1.1.1 Building Visual Basic Program

Before develop the program, thing must take consideration is the analysis to obtain the equivalent circuits and the transformer characteristic. The hand calculation was made to make sure there is no error in getting result. Then, the calculation was converted into Visual Basic programming language. All the symbols and equations must be defined.

$$\begin{aligned}
 A &= VP / VS \\
 ZP &= ESC / ISC \\
 RP &= PSC / (ISC ^ 2) \\
 XP &= (ZP ^ 2 - RP ^ 2) ^ 0.5 \\
 RM &= (ES ^ 2 * A ^ 2) / PM \\
 SM &= ES * IO \\
 QM &= (SM ^ 2 - PM ^ 2) ^ 0.5 \\
 XM &= (ES ^ 2 * A ^ 2) / QM \\
 \\
 SLreal &= SL * 1000 \\
 I2 &= SLreal / ES \\
 A &= VP / VS \\
 I1 &= I2 / A \\
 P\text{copper} &= (I1 ^ 2) * RP \\
 P\text{iron} &= PM \\
 P\text{losses} &= P\text{copper} - P\text{iron} \\
 PO &= SLreal * Q \\
 PI &= PO - P\text{losses} \\
 EFF &= (PO * 100) / PI
 \end{aligned}$$

3.1.1.2 Compile the Program

After the program was defined, compiling the program is necessary because the software will automatically display the error if the program has any. This is the

most importance step that to make sure the program is running without any mistake. If there is any error in defining the equation, the program cannot run properly.

3.1.1.3 Test the Program

The program now can be test with an input entered by user. This will make sure that the program can receive either the output data or the data that have been set in the program itself. If the program is not running properly, the program must be compiling again until the result is accurate.

3.1.2 The Graphical User Interface

After developing the program, the GUI will take place. The command that will be use in the GUI must be considered. The suitable command will make sure the program for interfacing with program that analyzes the data going to be easy.

3.1.2.1 Develop the GUI Using Visual Basic

For the GUI, the command buttons, text boxes, frames, and option buttons are used. All objects in the GUI must be defined according to the program that has been made. If the defining process has any error, the GUI would not run get the input and analyze the data.

```
S = Text1.Text  
VP = Text2.Text  
VS = Text3.Text  
PM = Text4.Text  
IO = Text5.Text  
ES = Text6.Text  
ESC = Text7.Text  
PSC = Text8.Text  
ISC = Text9.Text
```

```
Open "C:/VB.txt" For Output As #1
  Print #1, Text2.Text
  Print #1, Text3.Text
  Print #1, Text4.Text
  Print #1, Text6.Text
  Print #1, Text10.Text
  Print #1, Text19.Text
  Print #1, Text20.Text
Close #1

Open "C:/VB.txt" For Input As #1
  Input #1, variable1, variable2, variable3, variable4, variable5, variable6,
  variable7
  Text2.Text = variable1
  Text3.Text = variable2
  Text4.Text = variable3
  Text6.Text = variable4
  Text10.Text = variable5
  Text19.Text = variable6
  Text20.Text = variable7
Close #1
```

3.1.2.2 Displaying the Data in the GUI

The program was run and the results of the analysis were displayed in the text boxes. The limit of the graph was set for two cases which are 25 to 125 percent of full-load and 1 kVA to 300 MVA. The data must be defined properly to avoid any error in the analysis and misunderstanding by the user.

3.1.2.3 Test the GUI

The GUI was test and successfully developed.

CHAPTER 4

RESULT, DISCUSSION AND ANALYSIS

After completing the project, the result is so impressive where all expected results are achieved. The system now can analyze the single-phase transformer and view the efficiency of the transformer. The GUI that developed using Visual Basic is working properly. The program is ready to use by students for their learning purpose.

4.1 Result of the Visual Basic Program

The GUI has been developed by using Visual Basic 6. The first form show the transformer test and the equivalent circuits while the second show the transformer characteristic including the efficiency from 25 to 125 percent of full-load.

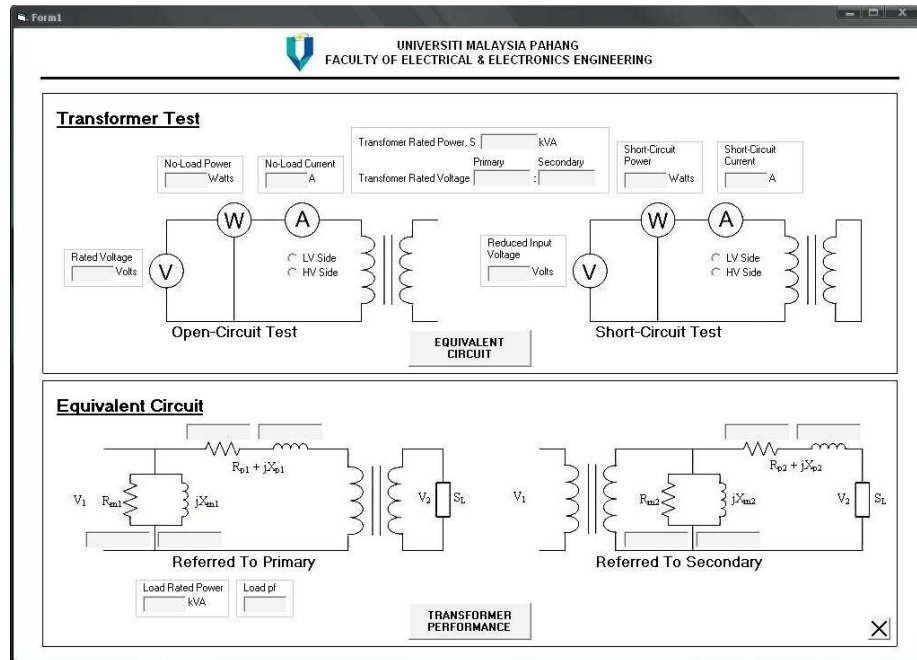


Figure 4.1 (a): Show the program main page where user prompted to enter the required data.

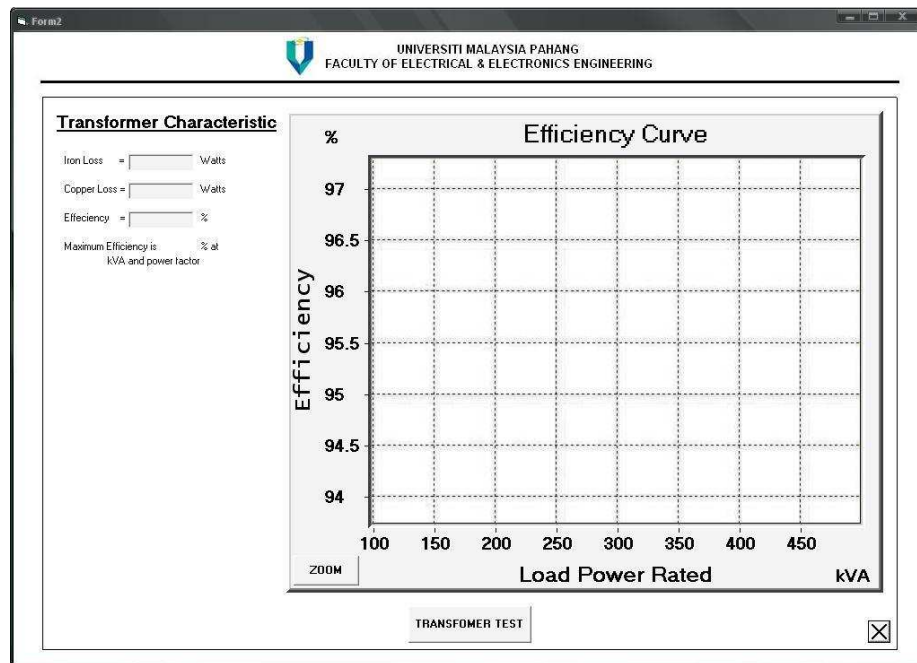


Figure 4.1 (b): Show the transformer efficiency and efficiency curve from 25 to 125% full load.

4.1.1 Transformer Tests

Using the program is not difficult. User need to fill the blank with the data from transformer tests and select the side which the test were conduct. There is an example for user and it can be used to compare the result with the simulation.

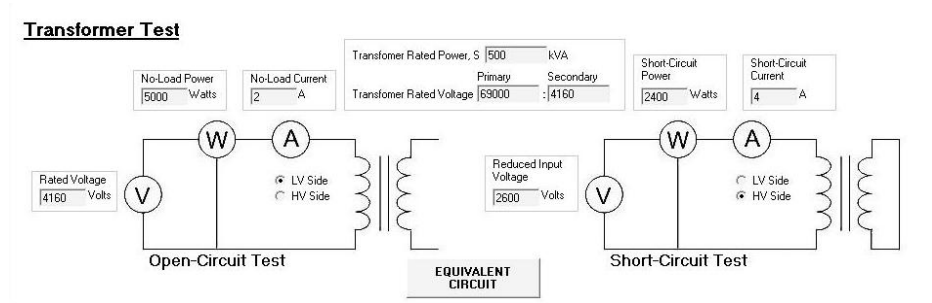


Figure 4.2: Data to be analyze.

Transformer rated power = 500 kVA
 Transformer rated voltage = 69000:4160 V

Open circuit test:-

Rated voltage = 4160 V
 No load power = 5000 W
 No load current = 2 A

Short circuit test:-

Reduced input voltage = 2600 V
 Short circuit power = 2400 W
 Short circuit current = 4 A

4.1.2 Equivalent Circuit of the Transformer

The program now analyzes the data and the view the result of the equivalent circuits that referred to the primary and secondary winding of the transformer. Below is a calculation to obtain the equivalent circuit.

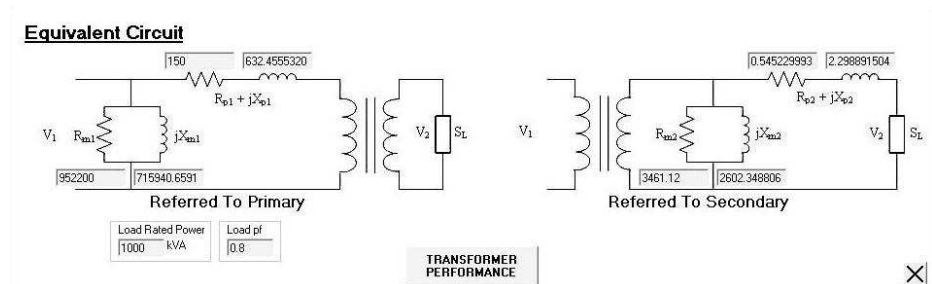


Figure 4.3: The equivalent circuits.

$$a = \frac{V_p}{V_s} = \frac{69000}{4160} = 16.587$$

$$Z_p = \frac{E_{sc}}{I_{sc}} = \frac{2600}{4} = 650 \Omega$$

$$R_p = \frac{P_{sc}}{I_{sc}^2} = \frac{2400}{4^2} = 150 \Omega$$

$$X_p = \sqrt{Z_p^2 - R_p^2} = \sqrt{650^2 - 150^2} = 632.456 \Omega$$

$$R_m = \frac{E_s^2(a^2)}{P_m} = \frac{4160^2(16.587^2)}{5000} = 952252.993 \Omega$$

$$S_m = E_s(I_o) = 4160(2) = 8320 \text{ VA}$$

$$Q_m = \sqrt{S_m^2 - P_m^2} = \sqrt{8320^2 - 5000^2} = 6649.992 \Omega$$

$$X_m = \frac{E_s^2(a^2)}{Q_m} = \frac{4160^2(16.587^2)}{6649.992} = 715980.555 \Omega$$

4.1.3 Transformer Efficiency

To obtain the efficiency of the transformer, user needs to enter the load specification that connected to the transformer terminal for an example; the 250kVA load with power factor 0.8 was connecting to the transformer. Before we continue the analysis, data need to be saved in text file so we can load the same data in second form where the efficiency of the transformer was viewed. It is one of the methods to transfer the data between forms. The result is show below in Figure 4.4.

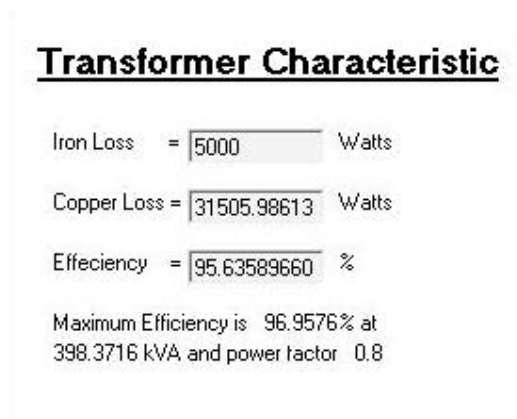


Figure 4.4: Transformer efficiency.

$$I_2 = \frac{S_L}{E_s} = \frac{250000}{4160} = 60.096 \text{ A}$$

$$I_1 = \frac{I_2}{a} = \frac{60.096}{16.587} = 3.623 \text{ A}$$

$$P_{copper} = I_1^2(R_p) = 3.623^2(150) = 1968.919 \text{ W}$$

$$P_{losses} = P_{copper} + P_{iron} = 1968.919 + 5000 = 6968.919 \text{ W}$$

$$P_o = S_L(p.f.) = 250000(0.8) = 200000 \text{ W}$$

$$P_i = P_o + P_{losses} = 200000 + 6968.919 = 206968.919 \text{ W}$$

$$\eta = \frac{P_o}{P_i}(100\%) = \frac{200000}{206968.919}(100\%) = 96.63\%$$

Maximum efficiency occur when $P_{copper} = P_{iron}$

$$I_1 = \sqrt{\frac{P_m}{R_p}} = \sqrt{\frac{5000}{150}} = 5.774 \text{ A}$$

$$I_2 = I_1(a) = 5.774(16.587) = 95.774 \text{ A}$$

$$S_L = I_2(E_s) = 95.774(4160) = 398419.84 \text{ VA}$$

$$P_o = S_L(p.f.) = 398419.84(0.8) = 318735.872 \text{ W}$$

$$P_i = P_o + (2P_m) = 318735.872 + 2(5000) = 328735.872 \text{ W}$$

$$\eta = \frac{P_o}{P_i}(100\%) = \frac{318735.872}{328735.872}(100\%) = 96.96\%$$

For display graph in Visual Basic, the Dyna Plot is used because it is more simple compared to other VB component's. Dyna Plot is a 32-bit chart control for technical and scientific applications for Windows. The installer comes with the samples on many types of graph.

The program now displays the efficiency curve of the transformer with the data from the above calculation. The program will set the limit of the graph so that the user can see the highest point of the efficiency. However, user can also zoom out the graph to view from 1 to 300 MVA of load.

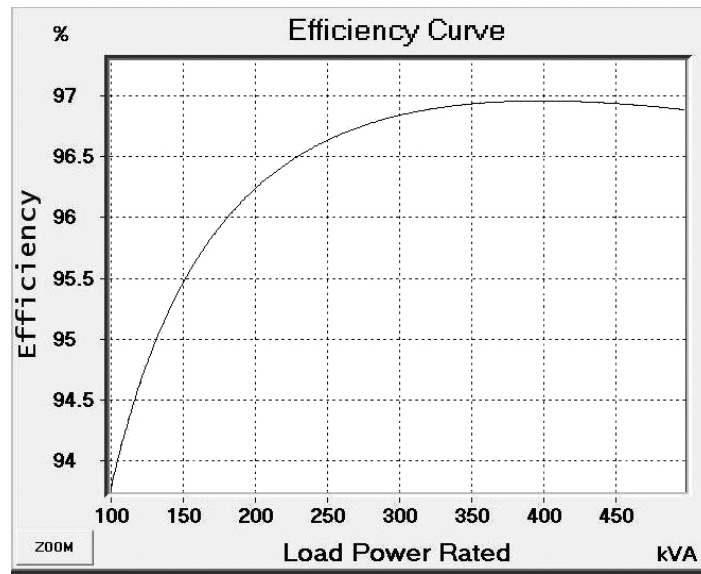


Figure 4.5 (a): Show the efficiency curve from 25 to 125 percent of full-load.

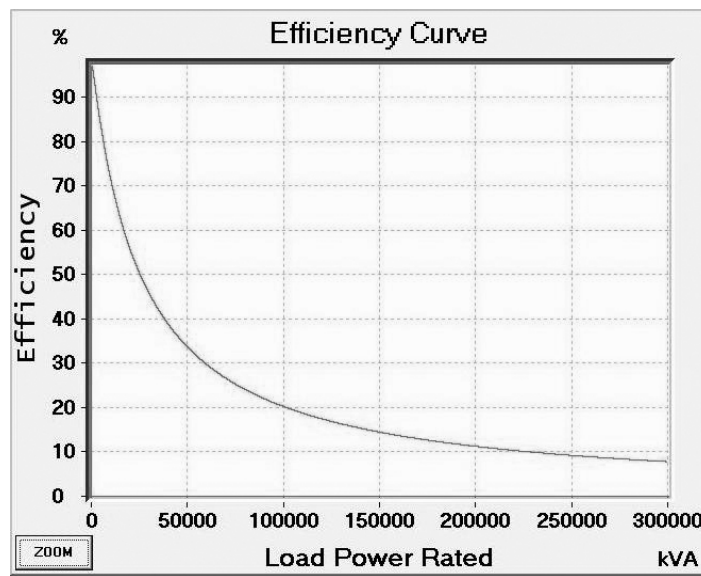


Figure 4.5 (b): Show the efficiency from 1 to 300MVA of load.

4.2 Overall of Project Operation

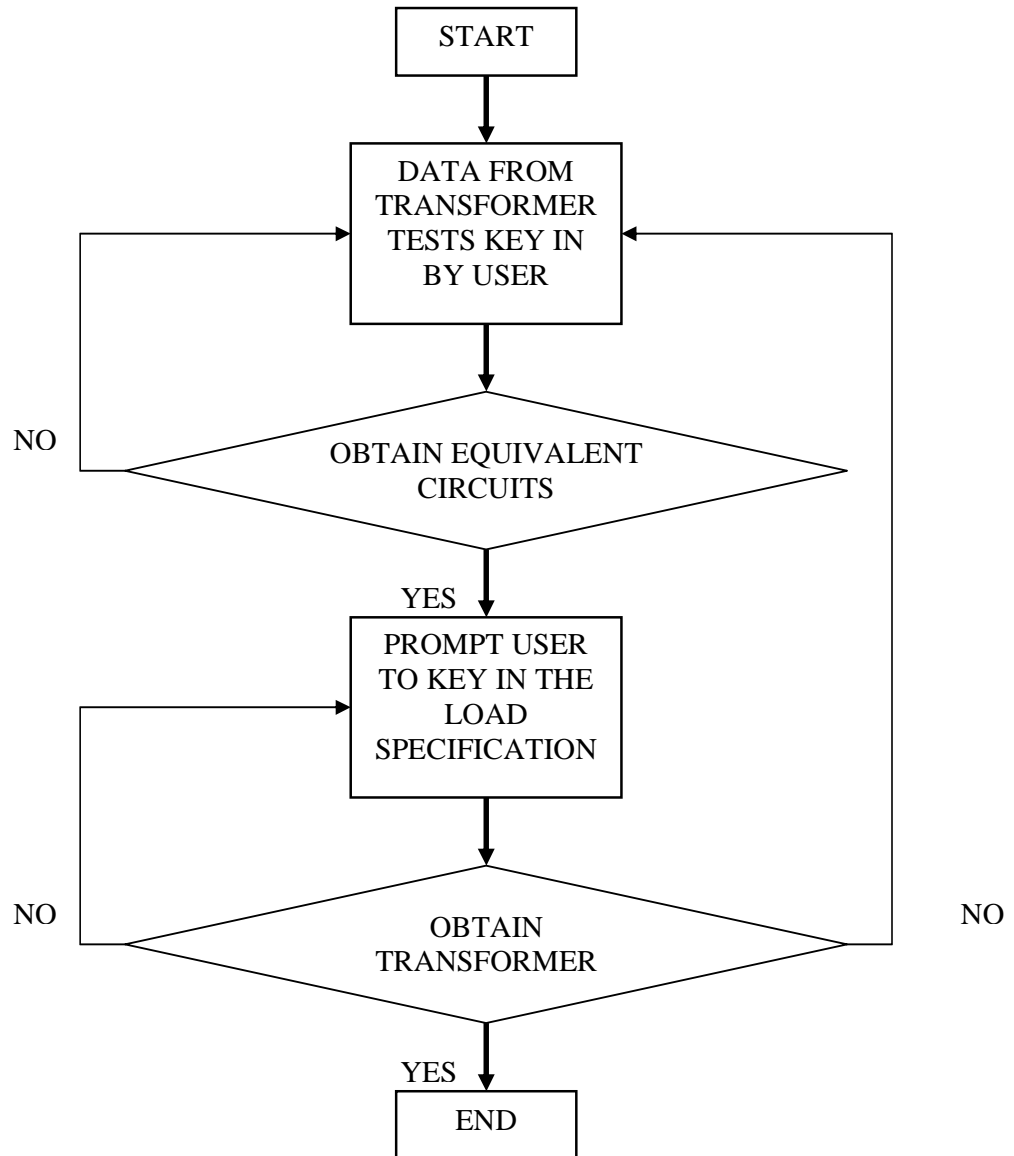


Figure 4.6: Block diagram of system flow

The first form show the transformer test and the equivalent circuits while the second show the transformer characteristic including the efficiency from 25 to 125 percent of full-load. Using the program is not difficult. User need to fill the blank with the data from transformer tests and select the side which the test were conduct. Then, user need to press the “EQUIVALENT CIRCUITS” button.

When the “EQUIVALENT CIRCUITS” button was pressed, the program now analyzes the data and the view the result of the equivalent circuits that referred to the primary and secondary winding of the transformer. User can change the transformer tests data for other analysis.

To obtain the efficiency of the transformer, user needs to enter the load specification that connected to the transformer terminal. Then, user need to press the “TRANSFORMER PERFORMANCE” button to get the efficiency of the transformer.

After the “TRANSFORMER PERFORMANCE” button was press, the program loads the second form and now displays the efficiency curve of the transformer with the data from the above calculation. The program will set the limit of the graph so that the user can see the highest point of the efficiency. However, user can also zoom out the graph by press the “ZOOM” button to view from 1 to 300 MVA of load. User also can press the “TRANSFORMER TEST” button to do another analysis back from the transformer test.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Summary of the Work

The simulation and development of a software package for Single Phase Transformer Analysis has been presented in this project. The development of the user friendly software package using Visual Basic is successfully completed after thorough and methodical research of the transformer analysis studies. Through the magnitude of the project, it is concluded that VB is a good platform to develop a user friendly software package to calculate and analyze the transformer. The designed GUI is actually complicated rather than the looks but it took efforts and hard work to do advance programming language. The objective of this project is to analyses the single phase transformer and generates the equivalent circuit comprising an ideal transformer and resistance and reactance is fully developed and also able to solve the characteristic such as efficiency of the transformer. The systems offer the suitable medium learning for electrical machine for students.

5.2 Recommendations For the Future Work

For the future development and enhancement, there are much more can do with this system to improve the system. Here is some enhancement of this project that should add in future:-

1. Upgrade the program for more function and not limited for single phase transformer only. There are many electrical machine can be analyze such as

induction machine, generator, even the fault analysis in the transmission system.

2. Upgrade the system including save function. The output from the transformer analysis can be future references if the calculation and the graph can be save into Microsoft Excel.
3. The basic feature of the software could improve by adding some valuable information and entertainment to the GUI. The layout could be smoothen or redesign to be more user friendly. Adding the instruction or program's manual together along with the program also can help first timer.

REFERENCES

1. Hadi Saadat (1999). *Power System Analysis*. WCB McGraw-Hill
2. Theodore Wildi (2006). *Electrical Machines, Drives, And Power System*, 6th Edition. Pearson, Prentice Hall
3. Jimmie J. Cathey, Syed A. Nasar (1997). *Basic Electrical Engineering*, 2nd Edition. WCB McGraw-Hill
4. Visual Basic URL
http://en.wikipedia.org/wiki/Visual_Basic
5. Visual Basic Tutorial URL
<http://www.vbtutor.net/lesson1.html>
6. Microsoft Visual Basic URL
<http://www.devdos.com/vb/wanttobe.shtml>

APPENDICES

APPENDIX A

Program Design

```
Private Sub Form_Load()  
    Frame8.Visible = False  
    Frame13.Visible = False  
End Sub
```

Program to select test cases

```
Private Sub Option1_Click()  
    Option4.Value = True  
End Sub
```

```
Private Sub Option2_Click()  
    Option3.Value = True  
End Sub
```

```
Private Sub Option3_Click()  
    Option2.Value = True  
End Sub
```

```
Private Sub Option4_Click()  
    Option1.Value = True  
End Sub
```

Declare the variable

```
Private Sub Command1_Click()  
  
    S = Text1.Text  
    VP = Text2.Text  
    VS = Text3.Text  
    PM = Text4.Text  
    IO = Text5.Text  
    ES = Text6.Text  
    ESC = Text7.Text  
    PSC = Text8.Text  
    ISC = Text9.Text
```

Begin the calculations

```
If Option1.Value = True Then  
    A = VP / VS
```



```

ZP = ESC / ISC
RP = PSC / (ISC ^ 2)
XP = (ZP ^ 2 - RP ^ 2) ^ 0.5
RM = (ES ^ 2 * A ^ 2) / PM
SM = ES * IO
QM = (SM ^ 2 - PM ^ 2) ^ 0.5
XM = (ES ^ 2 * A ^ 2) / QM
Text10.Text = RP
Text11.Text = XP
Text12.Text = RM
Text13.Text = XM
Text14.Text = RM / A ^ 2
Text15.Text = XM / A ^ 2
Text16.Text = RP / A ^ 2
Text17.Text = XP / A ^ 2
End If

If Option2.Value = True Then
    A = VP / VS
    ZP = ESC / ISC
    RP = PSC / (ISC ^ 2)
    XP = (ZP ^ 2 - RP ^ 2) ^ 0.5
    RM = (ES ^ 2 * A ^ 2) / PM
    SM = ES * IO
    QM = (SM ^ 2 - PM ^ 2) ^ 0.5
    XM = (ES ^ 2 * A ^ 2) / QM
    Text10.Text = RP / A ^ 2
    Text11.Text = XP / A ^ 2
    Text12.Text = RM / A ^ 2
    Text13.Text = XM / A ^ 2
    Text14.Text = RM
    Text15.Text = XM
    Text16.Text = RP
    Text17.Text = XP
End If

End Sub

```

Save the output in text form

```

Private Sub Command2_Click()

    Open "C:/VB.txt" For Output As #1
    Print #1, Text2.Text
    Print #1, Text3.Text
    Print #1, Text4.Text

```

```

    Print #1, Text6.Text
    Print #1, Text10.Text
    Print #1, Text19.Text
    Print #1, Text20.Text
    Close #1

```

```

    Form2.Show
    Form1.Hide

```

```
End Sub
```

```

Private Sub Command3_Click()
    Beep
    End
End Sub

```

Load the output to second form

```

Private Sub Form_Load()
    Text2.Visible = False
    Text3.Visible = False
    Text4.Visible = False
    Text6.Visible = False
    Text10.Visible = False
    Text19.Visible = False
    Text20.Visible = False

    Open "C:/VB.txt" For Input As #1
    Input #1, variable1, variable2, variable3, variable4, variable5, variable6, variable7
        Text2.Text = variable1
        Text3.Text = variable2
        Text4.Text = variable3
        Text6.Text = variable4
        Text10.Text = variable5
        Text19.Text = variable6
        Text20.Text = variable7
    Close #1

```

Begin the calculations

```

VP = Text2.Text
VS = Text3.Text
PM = Text4.Text
ES = Text6.Text
RP = Text10.Text
SL = Text19.Text
Q = Text20.Text

SLreal = SL * 1000
I2 = SLreal / ES

```

$A = VP / VS$
 $I1 = I2 / A$
 $P_{copper} = (I1^2) * RP$
 $P_{iron} = PM$
 $P_{losses} = P_{copper} - P_{iron}$
 $PO = SL_{real} * Q$
 $PI = PO - P_{losses}$
 $EFF = (PO * 100) / PI$

Text22.Text = Piron
Text23.Text = P_{copper}
Text24.Text = EFF

$I12 = (PM / RP)^{0.5}$
 $I22 = I12 * A$
 $SL_{real2} = I22 * ES$
 $SL2 = SL_{real2} / 1000$
 $PO2 = SL_{real2} * Q$
 $PI2 = PO2 - (2 * PM)$
 $EFF2 = (PO2 * 100) / PI2$

Text25.Text = EFF2
Text26.Text = SL2
Text27.Text = Q

$X1 = SL2 * 0.25$
 $X2 = SL2 * 1.25$

$SL3 = X1$
 $SL_{real3} = SL3 * 1000$
 $I23 = SL_{real3} / ES$
 $A = VP / VS$
 $I13 = I23 / A$
 $P_{copper3} = (I13^2) * RP$
 $P_{iron} = PM$
 $P_{losses3} = P_{copper3} - P_{iron}$
 $PO3 = SL_{real3} * Q$
 $PI3 = PO3 - P_{losses3}$
 $EFF3 = (PO3 * 100) / PI3$

$Y1 = EFF3$
 $Y2 = ((EFF2 - EFF3) / 10) - EFF2$

Set the graph axis and unit

DynaPlot1.Axes.XAxis.From = X1
DynaPlot1.Axes.XAxis.To = X2
DynaPlot1.Axes.YAxis.From = Y1
DynaPlot1.Axes.YAxis.To = Y2

```

DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_X).Text = "Load Power Rated"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_XUNIT).Text = "kVA"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_Y).Text = "Efficiency"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_YUNIT).Text = "%"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_TITLE).Text = "Efficiency
Curve"

```

Plot the graph

```

Const N = 200000
Dim y(N - 1)
Dim i As Long
Dim C As Curve

For i = 0 To N - 1
    y(i) = (i * Q * 100000) / ((i * Q * 1000) - -(PM) - -(((i * 1000) / (ES * A)) ^ 2)
    * RP))
Next
Set C = DynaPlot1.DataCurves.AddParametric("One", 0, 1, y, -1).Curve

```

End Sub

```

Private Sub Command1_Click()
    Beep
End
End Sub

```

```

Private Sub Command2_Click()
    Form1.Show
    Unload Me
End Sub

```

Zoom the graph

```

Private Sub Command3_Click()
    Dim X, W
    W = 0
    W = Text1.Text
    If W < 1 Then
        X = W - -1
        Text1.Text = X
    End If
    If W = 1 Then
        Text1.Text = 0
    End If
    If Text1.DataChanged Then GoTo bing:
bing:

```

```
Text2.Visible = False
Text3.Visible = False
Text4.Visible = False
Text6.Visible = False
Text10.Visible = False
Text19.Visible = False
Text20.Visible = False
```

```
Open "C:/VB.txt" For Input As #1
```

```
Input #1, variable1, variable2, variable3, variable4, variable5, variable6, variable7
```

```
Text2.Text = variable1
Text3.Text = variable2
Text4.Text = variable3
Text6.Text = variable4
Text10.Text = variable5
Text19.Text = variable6
Text20.Text = variable7
```

```
Close #1
```

```
VP = Text2.Text
VS = Text3.Text
PM = Text4.Text
ES = Text6.Text
RP = Text10.Text
SL = Text19.Text
Q = Text20.Text
```

```
SLreal = SL * 1000
I2 = SLreal / ES
A = VP / VS
I1 = I2 / A
Pcopper = (I1 ^ 2) * RP
Piron = PM
Plosses = Pcopper - Piron
PO = SLreal * Q
PI = PO - Plosses
EFF = (PO * 100) / PI
Text22.Text = Piron
Text23.Text = Pcopper
Text24.Text = EFF
```

```
I12 = (PM / RP) ^ 0.5
I22 = I12 * A
SLreal2 = I22 * ES
SL2 = SLreal2 / 1000
PO2 = SLreal2 * Q
PI2 = PO2 - (2 * PM)
EFF2 = (PO2 * 100) / PI2
Text25.Text = EFF2
Text26.Text = SL2
```

```

Text27.Text = Q
X1 = SL2 * 0.25
X2 = SL2 * 1.25

SL3 = X1
SLreal3 = SL3 * 1000
I23 = SLreal3 / ES
A = VP / VS
I13 = I23 / A
Pcopper3 = (I13 ^ 2) * RP
Piron = PM
Plosses3 = Pcopper3 - Piron
PO3 = SLreal3 * Q
PI3 = PO3 - Plosses3
EFF3 = (PO3 * 100) / PI3
Y1 = EFF3
Y2 = ((EFF2 - EFF3) / 10) - EFF2

If W = 0 Then
    DynaPlot1.Axes.XAxis.From = X1
    DynaPlot1.Axes.XAxis.To = X2
    DynaPlot1.Axes.YAxis.From = Y1
    DynaPlot1.Axes.YAxis.To = Y2
End If
If W = 1 Then
    DynaPlot1.Axes.XAxis.From = 0
    DynaPlot1.Axes.XAxis.To = 300000
    DynaPlot1.Axes.YAxis.From = 0
    DynaPlot1.Axes.YAxis.To = Y2
End If

DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_X).Text = "Load Power Rated"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_XUNIT).Text = "kVA"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_Y).Text = "Efficiency"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_YUNIT).Text = "%"
DynaPlot1.Chart.TextDescriptor(dpsTEXTDESC_TITLE).Text = "Efficiency
Curve"

Const N = 300000
Dim y(N - 1)
Dim i As Long
Dim C As Curve

For i = 0 To N - 1
    y(i) = (i * Q * 100000) / ((i * Q * 1000) - (PM) - (((i * 1000) / (ES * A)) ^ 2)
    * RP))
Next
Set C = DynaPlot1.DataCurves.AddParametric("One", 0, 1, y, -1).Curve
End Sub

```

APPENDIX B

Example Program for Calculation

```
Dim Op1, Op2
Dim DecimalFlag As Integer
Dim NumOps As Integer
Dim LastInput
Dim OpFlag
Dim TempReadout

Private Sub Cancel_Click()
    Readout = Format(0, "0.")
    Op1 = 0
    Op2 = 0
    Form_Load
End Sub

Private Sub CancelEntry_Click()
    Readout = Format(0, "0.")
    DecimalFlag = False
    LastInput = "CE"
End Sub

Private Sub Decimal_Click()
    If LastInput = "NEG" Then
        Readout = Format(0, "-0.")
    ElseIf LastInput <> "NUMS" Then
        Readout = Format(0, "0.")
    End If
    DecimalFlag = True
    LastInput = "NUMS"
End Sub

Private Sub Form_Load()
    DecimalFlag = False
    NumOps = 0
    LastInput = "NONE"
    OpFlag = " "
    Readout = Format(0, "0.")
End Sub

Private Sub Number_Click(Index As Integer)
    If LastInput <> "NUMS" Then
        Readout = Format(0, ".")
        DecimalFlag = False
    End If
    If DecimalFlag Then
        Readout = Readout + Number(Index).Caption
    End If
End Sub
```

```

Else
    Readout = Left(Readout, InStr(Readout, Format(0, ".")) - 1) +
    Number(Index).Caption + Format(0, ".")
End If
If LastInput = "NEG" Then Readout = "-" & Readout
    LastInput = "NUMS"
End If
End Sub

Private Sub Operator_Click(Index As Integer)
    TempReadout = Readout
    If LastInput = "NUMS" Then
        NumOps = NumOps + 1
    End If
    Select Case NumOps
    Case 0
        If Operator(Index).Caption = "-" And LastInput <> "NEG" Then
            Readout = "-" & Readout
            LastInput = "NEG"
        End If
    Case 1
        Op1 = Readout
        If Operator(Index).Caption = "-" And LastInput <> "NUMS" And OpFlag
            <> "=" Then
            Readout = "-"
            LastInput = "NEG"
        End If
    Case 2
        Op2 = TempReadout
        Select Case OpFlag
        Case "+"
            Op1 = CDb1(Op1) + CDb1(Op2)
        Case "-"
            Op1 = CDb1(Op1) - CDb1(Op2)
        Case "X"
            Op1 = CDb1(Op1) * CDb1(Op2)
        Case "/"
            If Op2 = 0 Then
                MsgBox "Can't divide by zero", 48, "Calculator"
            Else
                Op1 = CDb1(Op1) / CDb1(Op2)
            End If
        Case "="
            Op1 = CDb1(Op2)
        Case "%"
            Op1 = CDb1(Op1) * CDb1(Op2)
        End Select
        Readout = Op1
        NumOps = 1
    End Select

```



```
If LastInput <> "NEG" Then
    LastInput = "OPS"
    OpFlag = Operator(Index).Caption
End If
End Sub
```

```
Private Sub Percent_Click()
    Readout = Readout / 100
    LastInput = "Ops"
    OpFlag = "%"
    NumOps = NumOps + 1
    DecimalFlag = True
End Sub
```