DEVELOPMENT OF POWER SYSTEM ANALYSIS TOOLBOX USING MATLAB 7.1

MOHD AZZUAN BIN ABD AZIZ

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DEVELOPMENT OF POWER SYSTEM ANALYSIS TOOLBOX USING MATLAB 7.1**

SESI PENGAJIAN: **2008/2009**

Saya **MOHD AZZUAN BIN ABD AZIZ (860516-02-5197)**
(HURUF BESAR)

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

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
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 128, PEKAN KODIANG
06100, KODIANG
KEDAH DARUL AMAN

NORHAFIDZAH BT MOHD SAAD
(Nama Penyelia)

Tarikh: **12 NOVEMBER 2008**

Tarikh: : **12 NOVEMBER 2008**

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 (Power Systems)"

Signature : ____

Name : NORHAFIDZAH BT MOHD SAAD

Date : 17 NOVEMBER 2008

TABLE OF CONTENT

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

Signature : ______________________________

Author : MOHD AZZUAN BIN ABD AZIZ

Date : 17 NOVEMBER 2008

ACKNOWLEDGEMENT

Firstly, I am grateful to the almighty ALLAH s.w.t. for giving me this opportunity to accomplish my final year project.

Secondly, I wish to express my deep gratitude to my supervisor, Mrs Norhafidzah bt Mohd Saad for all her valuable guidance, assistance and support through this work. Without her continuous support, this thesis would not been the same as presented here.

Besides, I am also thankful to my parents for giving me encouragement and moral support while doing this thesis.

Lastly, I want to say thank you to all my friends who are willing help me especially my friends who are supervised by Pn. Norhafidzah bt Mohd Saad, thank you very much for everything.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

POWER SYSTEM ANALYSIS EDUCATIONAL TOOLBOX USING MATLAB 7.1

MOHD AZZUAN BIN ABD AZIZ

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

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2008

ABSTRACT

This Power System Analysis toolbox by using MATLAB GUI has been developed by the author to assist in typical Power System Analysis. The one of the objectives of this project is to develop an educational toolbox for Electrical Power System students and lecturers in order to solve some of Power System problems such as *Power Flow Analysis*, *Fault Analysis*, *Optimal Dispatch of Power Generation*, and lastly is, *Steady State and Transient Stability Analysis*. All this kinds of problems consists of various methods of mathematical calculation which is difficult to perform by using manual calculation (formula and calculator). The existence of this educational toolbox will help the user to calculate the calculation become more faster and easier. This educational toolbox was developed by using MATLAB 7.1 software (M-File and Graphical User Interface). MATLAB, with its extensive numerical resources, can be used to obtain numerical solutions that involve various types of vector-matrix operations. This Power System educational toolbox allows the user especially students to analyze and design power systems without having a lot of calculation.

CHAPTER 1

INTRODUCTION

1.1 Overview

Power System Analysis is an analysis that is so important nowadays. It is not only important in economic scheduling, but also necessary for planning and operation for a system. Based on that, in recently years, there are many researches, new developments and analysis was introduced to people in order to mitigate the problems that involving Power System Analysis such as Load Flow Analysis, Fault Analysis, Stability Analysis and Optimal Dispatch on Power Generation.

i) *Load Flow Analysis* is important to analyze any planning for power system improvement under steady state conditions such as to build new power generation capacity, new transmission lines in the case of additional or increasing of loads, to plan and design the future expansion of power systems as well as in determining the best operation of existing systems.

ii) *Fault Analysis* is important to determine the magnitude of voltages and line currents during the occurrence of various types of fault.

iii) *Stability Analysis* is necessary for reliable operation of power systems to keep synchronism after minor and major disturbances.

iv) *Optimal Dispatch* is to find real and reactive power to power plants to meet load demand as well as minimize the operation cost.

All the analysis discussed above is an importance tool involving numerical analysis that applied to a power system. In this analysis, there is no known analytical method to solve the problem because it depends on iterative technique. Iterative technique is one of the analysis that using a lot of mathematical calculations which takes a lot of times to perform by hand. So, to solve the problems, the development of this toolbox based on MATLAB 7.1 with Graphical User Interface (GUI) will help the analysis become quick and easy.

Over the past decade, a few versions of educational software packages using advanced programming languages such as C, Borland C++, Pascal or Fortran have been developed for power engineering curriculums. But, the author chose MATLAB 7.1 with GUI to develop this software since most of the students are familiar with MATLAB. In addition to that, MATLAB is a matrix-based software package, which makes it ideal for Power System Analysis. MATLAB, with its extensive numerical resources, can be used to obtain numerical solutions that involve various types of vector-matrix operations.

1.2 Objectives

The main objectives that the author wants to be achieved in this project are:

i) To develop an educational toolbox in order to solve Power System Analysis problems.
ii) To obtain simulation and analysis by using MATLAB GUI

1.3 Scope of Project

In this project, the author will focused on:

i) Study the theory of Power System Analysis that involves Load Flow Analysis, Fault Analysis, Stability Analysis and Optimal Dispatch of Power Generation.
ii) This project will concentrates on MATLAB 7.1 programming with Graphical User Interface (GUI).
iii) To perform simulation of Load Flow Analysis, Fault Analysis, Stability Analysis and Optimal Dispatch of Power Generation.

CHAPTER 2

LITERATURE REVIEW

The literature review will be divided into two parts. The first one is, the author will discuss on the theory of Load Flow, Fault Analysis, Stability Analysis and Optimal Dispatch of Power Generation meanwhile for second one is, on Graphical User Interface (GUI) that build MATLAB Software.

2.1 Load Flow Analysis

Power flow studies are the backbone for power system analysis and design. They are necessary for planning, operation, economic scheduling and exchange of power between utilities. Power flow analysis is required for many other analyses such as fault analysis, transient stability studies and contingency studies [1].

There are THREE methods that can be used to solve power flow analysis. The methods are Gauss-Seidel Method, Newton-Raphson method, and Fast Decoupled method.

i) Gauss-Seidel Method

Gauss-Seidel method is a technique used to solve a linear system of equations. The method is named after the German mathematicians Carl Friedrich Gauss and Philipp Ludwig von Seidel. The method is an improved version of the Jacobi method. It is defined on matrices with non-zero diagonals, but convergence is only guaranteed if the matrix is either diagonally dominant or symmetric and positive definite [2].

ii) Newton-Raphson method

In numerical analysis, Newton's method (also known as the Newton–Raphson method or the Newton–Fourier method) is perhaps the best known method for finding successively better approximations to the zeros (or roots) of a real-valued function. Newton's method can often converge remarkably quickly, especially if the iteration begins "sufficiently near" the desired root. Just how near "sufficiently near" needs to be and just how quickly "remarkably quickly" can be depends on the problem, as is discussed in detail below. Unfortunately, far from the desired root, Newton's method can easily lead an unwary user astray with little warning. Thus, good implementations of the method embed it in a routine that also detects and perhaps overcomes possible convergence failures [3].

The Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function. Its importance lies in the fact that it represents the best linear approximation to a differentiable function near a given point. In this sense, the Jacobian is akin to a derivative of a multivariate function. For $n > 1$, the derivative of a numerical function must be matrix-valued, or a partial derivative [4].

$$\underbrace{\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \cdots & \frac{\partial P_2}{\partial \delta_n} & |V_2|\frac{\partial P_2}{\partial |V_2|} & \cdots & |V_n|\frac{\partial P_2}{\partial |V_n|} \\ \vdots & J_{11} & \vdots & \vdots & J_{12} & \vdots \\ \frac{\partial P_n}{\partial \delta_2} & \cdots & \frac{\partial P_n}{\partial \delta_n} & |V_2|\frac{\partial P_n}{\partial |V_2|} & \cdots & |V_n|\frac{\partial P_n}{\partial |V_n|} \\ \hline \frac{\partial Q_2}{\partial \delta_2} & \cdots & \frac{\partial Q_2}{\partial \delta_n} & |V_2|\frac{\partial Q_2}{\partial |V_2|} & \cdots & |V_n|\frac{\partial Q_2}{\partial |V_n|} \\ \vdots & J_{21} & \vdots & \vdots & J_{22} & \vdots \\ \frac{\partial Q_n}{\partial \delta_2} & \cdots & \frac{\partial Q_n}{\partial \delta_n} & |V_2|\frac{\partial Q_n}{\partial |V_2|} & \cdots & |V_n|\frac{\partial Q_n}{\partial |V_n|} \end{bmatrix}}_{JACOBIAN\ \ MATRIX} \underbrace{\begin{bmatrix} \Delta\delta_2 \\ \vdots \\ \Delta\delta_n \\ \hline \frac{\Delta|V_2|}{|V_2|} \\ \vdots \\ \frac{\Delta|V_n|}{|V_n|} \end{bmatrix}}_{CORRECTION\ \ S} = \underbrace{\begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_n \\ \hline \Delta Q_2 \\ \vdots \\ \Delta Q_n \end{bmatrix}}_{MISMATCHES} \qquad (2.1)$$

iii) Fast decoupled method

The Fast decoupled power flow solution requires more iterations than the Newton-Raphson method, but requires considerably less time per iteration and a power flow solution is obtained rapidly. This technique is very useful in contingency analysis where numerous outages are to be simulated or a power flow solution is required for on-line control [1].

For large scale power system, usually the transmission lines have a very high X/R ratio. For such a system, real power changes ΔP are less sensitive to changes in voltage magnitude and are most sensitive to changes in phase angle $\Delta\delta$. Similarly, reactive power is less sensitive to changes in angle and most sensitive on changes in voltage magnitude. Incorporate of these approximations into the Jacobian matrix in Newton-Raphson power flow solution makes the elements of the submatrices J_{12} and J_{21} zero [5].

We are then left with two separated systems of equations

$$\begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \cdots & \frac{\partial P_2}{\partial \delta_n} \\ \vdots & J_{11} & \vdots \\ \frac{\partial P_n}{\partial \delta_2} & \cdots & \frac{\partial P_n}{\partial \delta_n} \end{bmatrix} \begin{bmatrix} \Delta \delta_2 \\ \vdots \\ \Delta \delta_n \end{bmatrix} = \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_n \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} |V_2| \frac{\partial Q_2}{\partial |V_2|} & \cdots & |V_n| \frac{\partial Q_2}{\partial |V_n|} \\ \vdots & J_{22} & \vdots \\ |V_2| \frac{\partial Q_n}{\partial |V_2|} & \cdots & |V_n| \frac{\partial Q_n}{\partial |V_n|} \end{bmatrix} \begin{bmatrix} \frac{\Delta |V_2|}{|V_2|} \\ \vdots \\ \frac{\Delta |V_n|}{|V_n|} \end{bmatrix} = \begin{bmatrix} \Delta Q_2 \\ \vdots \\ \Delta Q_n \end{bmatrix} \tag{2.3}$$

2.2 Fault Analysis

There are two types of fault that is occur in power system network which are [1]

a) Three phase balanced faults
b) Unbalance faults

Different types of unbalanced faults are [5]:

a) Single line to ground faults
b) Line to line faults
c) Double line to ground faults

The types of fault can be seen in Figure 2.1 below [1]

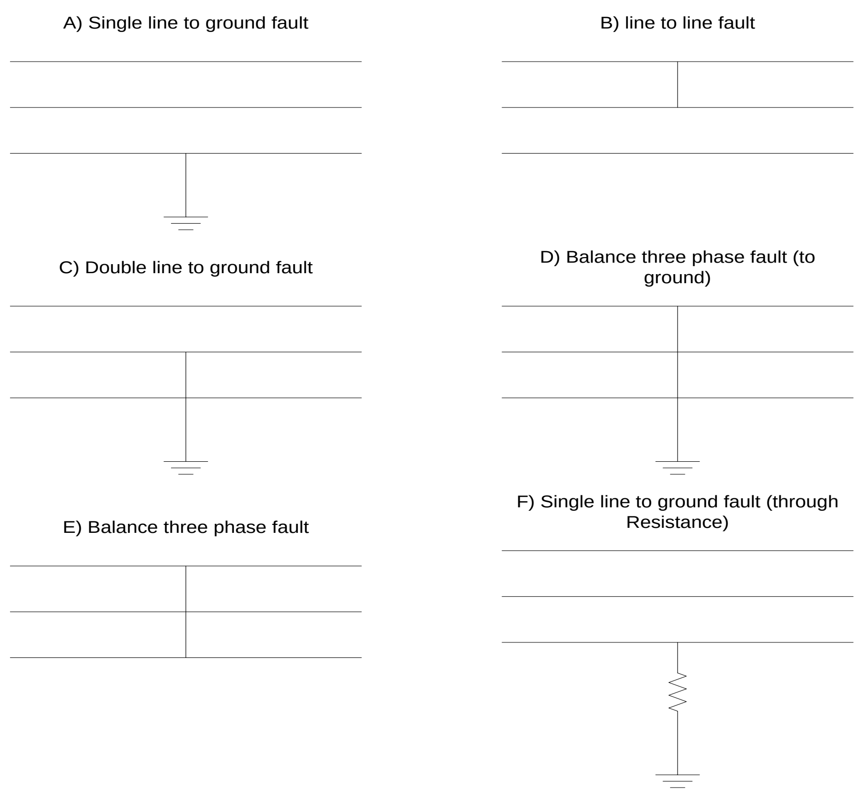


Figure 2.1 Types of Fault

Fault studies are very important in power system analysis. The problem consists of determining bus voltage and line currents during various types of faults. The information gained from fault studies are used for [1]:

a) Proper selection of circuit breaker.
b) Determination of relay setting and coordination which control the circuit breakers.
c) Select and set phase relays (for 3 phase balance fault) and ground relays (for line to ground fault).
d) Obtain the rating of protective switchgears

2.2.1 Balanced Three Phase Fault

This type of fault is defined as the simultaneous short circuit across all three phases. It occurs infrequently, but it is the most severy type of fault encountered. Because the network is balanced, it is solved on a per phase basis. The other phases carry identical current except for the phase shift [1].

2.2.2 Unbalanced Fault

The Figure 2.2, Figure 2.3, and Figure 2.4 below will illustrate about single line to ground fault, line to line fault, and double line to ground fault [1].

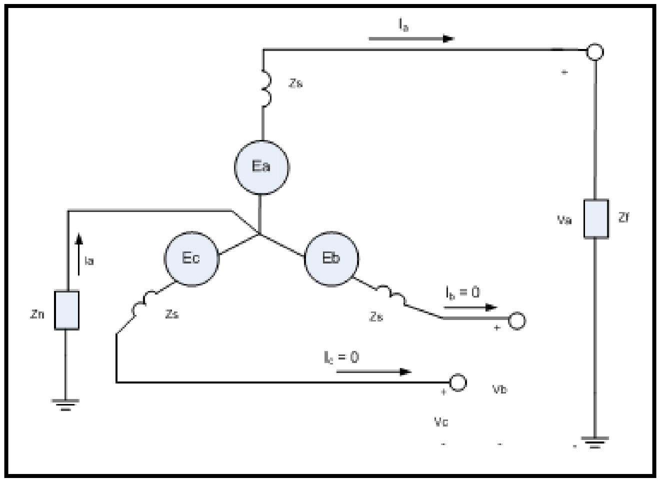


Figure 2.2 Single Line to Ground Fault

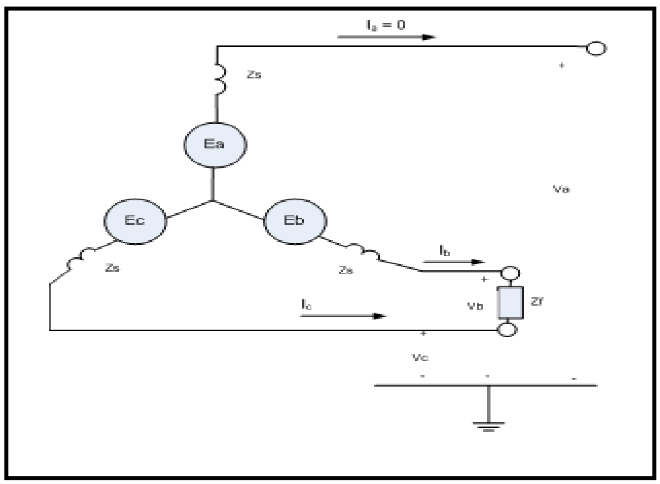


Figure 2.3 Line to Line Fault

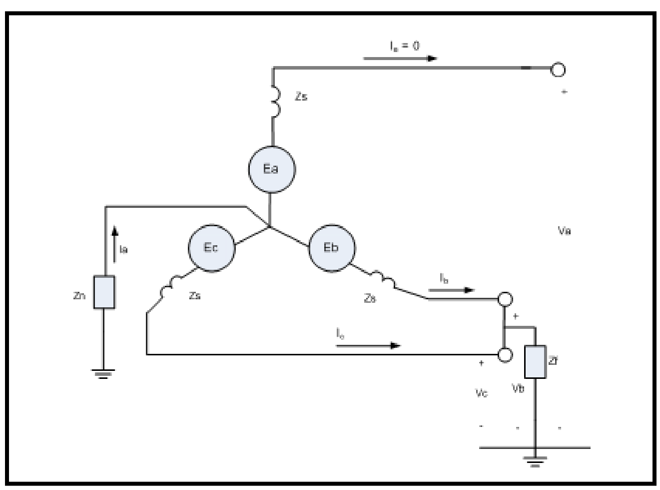


Figure 2.4 Double Line to Ground Fault

2.3.1 Stability Analysis

Stability is the ability of the power system to retain stable in normal operation after having endured some form of disturbance. Stability is conducted at planning level when new generating and transmitting facilities are developed. The studies are needed in determining the relaying system needed, critical fault clearing time of circuit breaker, critical clearing angle, auto reclosing time t_{cr}, voltage level and transfer capability between system. When the power system loss stability, the machines will lose synchronization and it will no longer working at synchronous speed. This will lead to power, voltage and current to oscillate drastically. It can cause damage to the loads which receive electric supply from the instable system [1].

2.3.1 Steady-State Stability

Steady-State Stability is the ability of a system to remain synchronism after small and slow disturbances [5].

2.3.2 Transient State Stability

Transient State Stability is the ability of the power system to maintain in stability after large, major and sudden disturbances. For example are, occurrence of faults, sudden load changes, loss of generating unit, line switching. The transient stability studies involve the determination of whether or not synchronism is maintained after the machine has been subjected to severe disturbance [1]. Types of disturbances [5]:

i) Sudden application of load/sudden load changing
ii) Loss of generation
iii) Fault on the system

2.3.3 Equal Area Criterion

A method known as equal area criterion can be used for a quick prediction of stability after the machine has been subjected to severe disturbance. For example is sudden change of load.

Lets consider the machine operating at equilibrium point, δ_0 , with mechanical power equal to electrical power, or $P_{m0}=P_{e0}$, as shown in Figure 2.5 [1].

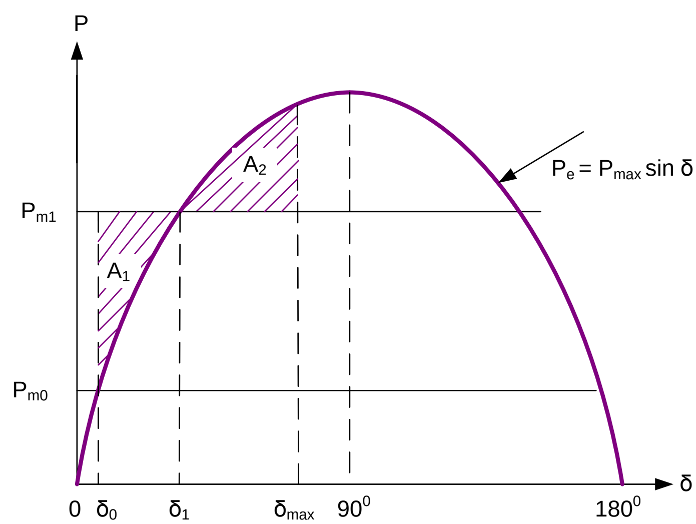


Figure 2.5 Equal area criterion: sudden change of load

Consider a sudden increase in input power (mechanical power) from P_{m0} to P_{m1}. Since $P_{m1} > P_{e0}$, the accelerating power of the rotor is positive and the power angle δ increases. The excess energy stored in the rotor during the initial acceleration is

$$\int_{\delta_0}^{\delta_1} P_{m1} - P_e \; d\delta = \text{area } A_1 \tag{2.4}$$

With increase in δ, the electrical power is increases, and when $\delta = \delta_1$, electrical power is equal to the new power input, P_{m1}. At this point, the accelerating power is zero but the rotor still running above synchronous speed. Hence, δ and electrical power P_e will continue to increase. Now $P_{m1} < P_e$, causing the rotor is decelerating towards synchronous speed until $\delta = \delta_{max}$. The energy given up by rotor as it decelerates back towards synchronous speed is

$$\int_{\delta_0}^{\delta_{max}} (P_e - P_{m1})\, d\delta = \text{area } A_2 \quad (2.5)$$

The power system maintain its stability if |Area A1 = Area A2|. This is known as the equal area criterion. The rotor angle will then oscillate back and forth between δ0 and δmax at its natural frequency [5].

At $A_1 = A_2$,

$$A_1 - A_2 = \int_{\delta_0}^{\delta_{max}} (P_{m1} - P_e)\, d\delta = 0 \quad (2.6)$$

The critical clearing angle, δc, is reached when δmax, is at intersection of Pm and Pe. The circuit breaker must open the faulted line before δ reached the critical clearing angle, δc. The time for power angle reach the critical clearing angle is called critical clearing time [5].

Critical clearing angle, δc is give by:

$$\cos \delta_c = \frac{P_m}{P_{max}} (\delta_{max} - \delta_o) + \cos \delta_{max} \quad (2.7)$$

Critical clearing time, tc is give by

$$t_c = \sqrt{\frac{2H (\delta_c - \delta_0)}{\pi f_0 P_m}} \quad (2.8)$$

2.4 Optimal Dispatch of Power Generation

Power Flow solution provided the voltage phase angle and the reactive power generation. In the practical power system, power plants are not at the same distance from the centre of load and their fuel costs are different. Also, under normal operating condition the generation capacity is more than the total load demand and losses. Thus there are many option for scheduling generation. In an interconnected power system, the objective is to find the real and reactive power scheduling of each power plant in such a way as to minimize the operating cost. This means that generator's real and reactive power are allowed to vary within certain limits so as to meet a particular load demand with minimum fuel cost. This is called the Optimal Power Flow (OPF) problem [5].

The OPF is used to optimize the power flow solution large scale power system. This is done by minimizing selected objective functions while maintaining an acceptable system performance in terms of generator capability limits and the output of the compensating devices. The objective function, also known as cost function, may present economic costs, and system security [5] .

Lets say, there is a system like Figure 2.6 that consists of N thermal-generating units connected to a single bus-bar serving a received electrical load P_{load}. The input to each unit, shown as F_i, represents the cost rate of the unit. The total cost rate of this system is, of course, the sum of the costs of each of the individual units. The essential constraint on the operation of this system is that the sum of the output powers must equal to the load demand [6].

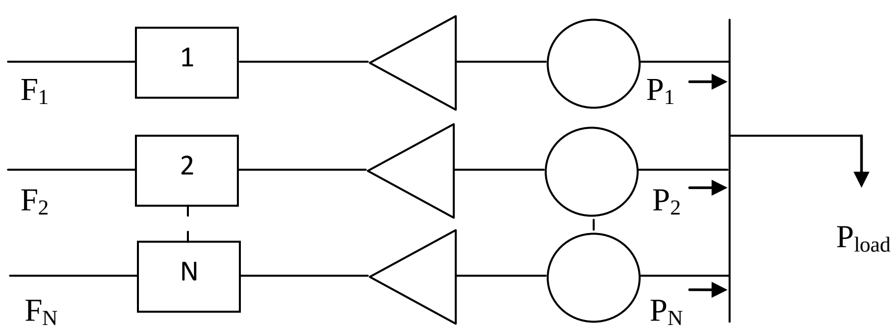


Figure 2.6 N thermal units committed to serve a load of P_{load}

2.4.1 Economic Dispatch with Generator Limits

The output power for any generator should not be exceed its rating or be below the value for stable boiler operation. Thus the generators must have a minimum and maximum real power output limits. But the problem is to find the real power generation for each plant such that cost are minimized, subject to Meeting load demand (equality constraints) and Constrained by the generator limits (inequality constraints) [5].

The Kuhn-Tucker condition

$$\begin{aligned} dC_i/dP_i &= \lambda &\leftarrow \quad & P_{i(\min)} < P_i < P_{i(\max)} \\ dC_i/dP_i &\le \lambda &\leftarrow \quad & P_i = P_{i(\max)} \\ dC_i/dP_i &\ge \lambda &\leftarrow \quad & P_i = P_{i(\min)} \end{aligned} \tag{2.9}$$

2.4.2 Economic Dispatch including Losses

For large interconnected system where power is transmitted over long distances with low load density areas, transmission line losses are a major factor and affect the optimum dispatch of generation. One common practice for including the effect of transmission losses is to express the total transmission loss as a quadratic function of the generator power outputs [5]. The simplest quadratic form is

$$P_L = \sum_{i=1}^{n_{gen}} \sum_{j=1}^{n_{gen}} P_i B_{ij} P_j \tag{2.10}$$

If Kron's loss formula be used, the equation is

$$P_L = \sum_{i=1}^{n_{gen}} \sum_{j=1}^{n_{gen}} P_i B_{ij} P_j + \sum_{j=1}^{n_{gen}} B_{0j} P_j + B_{00} \tag{2.11}$$

2.5 GRAPHICAL USER INTERFACE

Graphical User Interface (GUI) is a type of user interface which allows people to interact with a computer and computer-controlled devices. As opposed to traditional interface, it presents graphical icons, visual indicators or special graphical elements called "widgets". Often the icons are used in conjunction with text, labels or text navigation to fully represent the information and actions available to a user. But instead of offering only text menus, or requiring typed commands, the actions are usually performed through direct manipulation of the graphical elements [7].

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance and with intuitive controls like pushbuttons, list boxes, sliders, menus, and so forth. The GUI should behave in an understandable and predictable manner, so that a user knows what to expect when he or she performs an action. For example, when a mouse click occurs on a pushbutton, the GUI should initiate the action described on the label of the button. This chapter introduces the basic elements of the MATLAB GUIs. The chapter does not contain a complete description of components or GUI features, but it does provide the basics required to create functional GUIs for your program [8].

A graphical user interface provides the user with a familiar environment in which to work. This environment contains pushbuttons, toggle buttons, lists, menus, text boxes, and so forth, all of which are already familiar to the user, so that he or she can concentrate on using the application rather than on the mechanics involved in doing things. However, GUIs are harder for the programmer because a GUI-based program must be prepared for mouse clicks (or possibly keyboard input) for any GUI element at any time. Such inputs are known as events, and a program that responds to events is said to be event driven. The three principal elements required to create a MATLAB Graphical User Interface are [8]:

i) Components.

Each item on a MATLAB GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function uicontrol, and menus are created by the functions uimenu and uicontextmenu. Axes, which are used to display graphical data, are created by the function axes.

ii) Figures.

The components of a GUI must be arranged within a figure, which is window on the computer screen. In the past, figures have been created automatically whenever we have plotted data. However, empty figures can be created with the function figure and can be used to hold any combination of components.

iii) Callbacks.

Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a call back. There must be a callback to implement the function of each graphical component on the GUI. The basic GUI elements are summarized in Table 2.1.

Table 2.1 : Basic GUI Components

Table 10.1 Some Basic GUI Components

Element	Created By	Description
Graphical Controls		
Pushbutton	`uicontrol`	A graphical component that implements a pushbutton. It triggers a callback when clicked with a mouse.
Toggle button	`uicontrol`	A graphical component that implements a toggle button. A toggle button is either "on" or "off," and it changes state each time that it is clicked. Each mouse button click also triggers a callback.
Radio button	`uicontrol`	A radio button is a type of toggle button that appears as a small circle with a dot in the middle when it is "on." Groups of radio buttons are used to implement mutually exclusive choices. Each mouse click on a radio button triggers a callback.
Check box	`uicontrol`	A check box is a type of toggle button that appears as a small square with a check mark in it when it is "on." Each mouse click on a check box triggers a callback.
Edit box	`uicontrol`	An edit box displays a text string and allows the user to modify the information displayed. A callback is triggered when the user presses the `Enter` key.
List box	`uicontrol`	A list box is a graphical control that displays a series of text strings. A user can select one of the text strings by single- or double-clicking on it. A callback is triggered when the user selects a string.
Popup menus	`uicontrol`	A popup menu is a graphical control that displays a series of text strings in response to a mouse click. When the popup menu is not clicked on, only the currently selected string is visible.
Slider	`uicontrol`	A slider is a graphical control to adjust a value in a smooth, continuous fashion by dragging the control with a mouse. Each slider change triggers a callback.
Static Elements		
Frame	`uicontrol`	Creates a frame, which is a rectangular box within a figure. Frames are used to group sets of controls together. Frames never trigger callbacks.
Text field	`uicontrol`	Creates a label, which is a text string located at a point on the figure. Text fields never trigger callbacks.
Menus and Axes		
Menu items	`uimenu`	Creates a menu item. Menu items trigger a callback when a mouse button is released over them.
Context menus	`uicontextmenu`	Creates a context menu, which is a menu that appears over a graphical object when a user right-clicks the mouse on that object.
Axes	`axes`	Creates a new set of axes to display data on. Axes never trigger callbacks.

CHAPTER 3

METHODOLOGY

3.1 Introduction

On this chapter, it will discuss on the methodology of this project. It will describe on how this project is conducted and also the steps that will be followed in order to complete the project. This methodology will be separated into TWO parts, which are flow chart of this project and development of GUI software.

3.2 Flow Chart of Project

The flow chart of the project can be seen in Figure 3.1

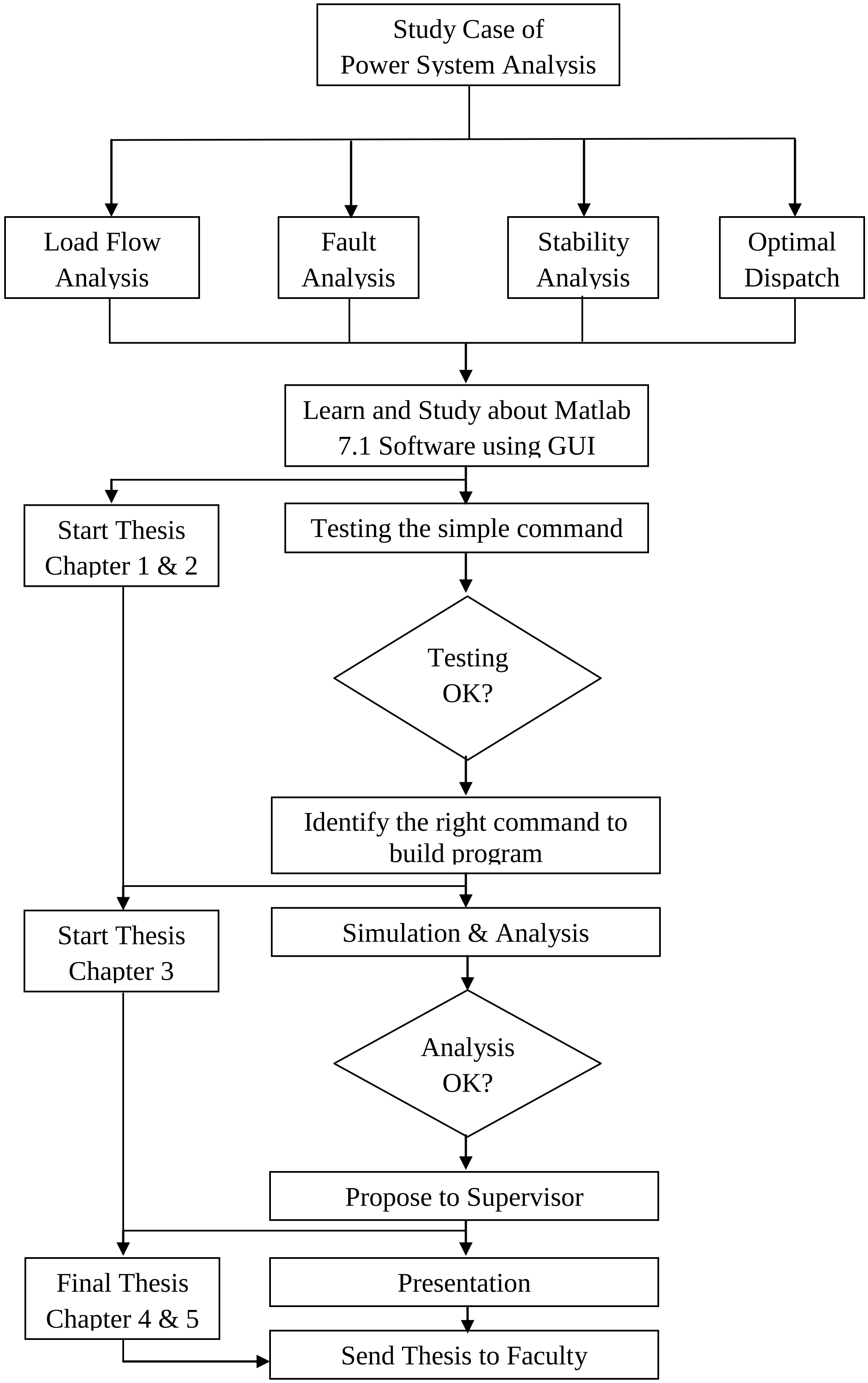


Figure 3.1 Flow Chart Project

3.3 GUI Development

GUIDE, the MATLAB Graphical User Interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). Usually, GUI will give the user to design their own layout of a program. In this section, we will discuss about the development in GUI on how this project be conducted.

3.3.1 Creating Graphical User Interfaces (GUIs)

To start GUIDE, enter *guide* at Matlab prompt. The display of GUIDE Quick Start dialog, is shown in Figure 3.2

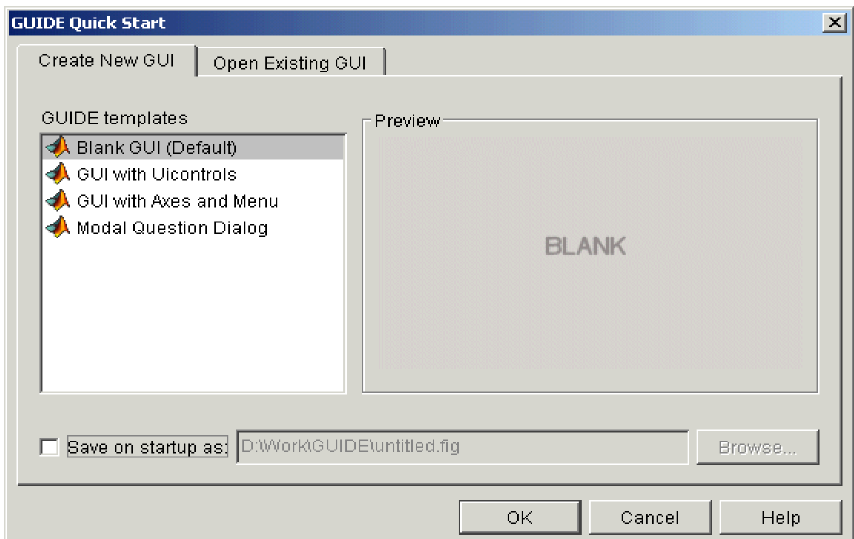


Figure 3.2 Main Page of GUI

From the Quick Start dialog, user can create a new GUI from one of the GUIDE templates or open an existing GUI. The Create New GUI part will be used to create a new GUI program and after select the option, click OK. The result should be appear as shown in Figure 3.3.

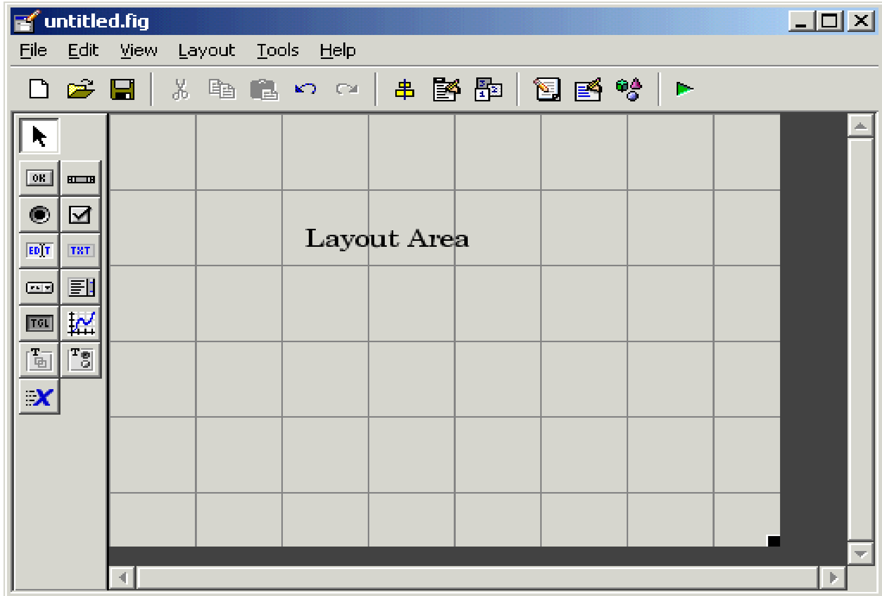


Figure 3.3 Layout Area of GUI

The Open Existing GUI is used to callback the previous project that we have saved before.

3.3.2 Layout the GUI

Using the GUIDE Layout Editor, the user can lay out a GUI easily by clicking and dragging GUI component such as panels, buttons, text fields, sliders, menus, and so on into the layout area. All of this component palette have their own function in GUI.

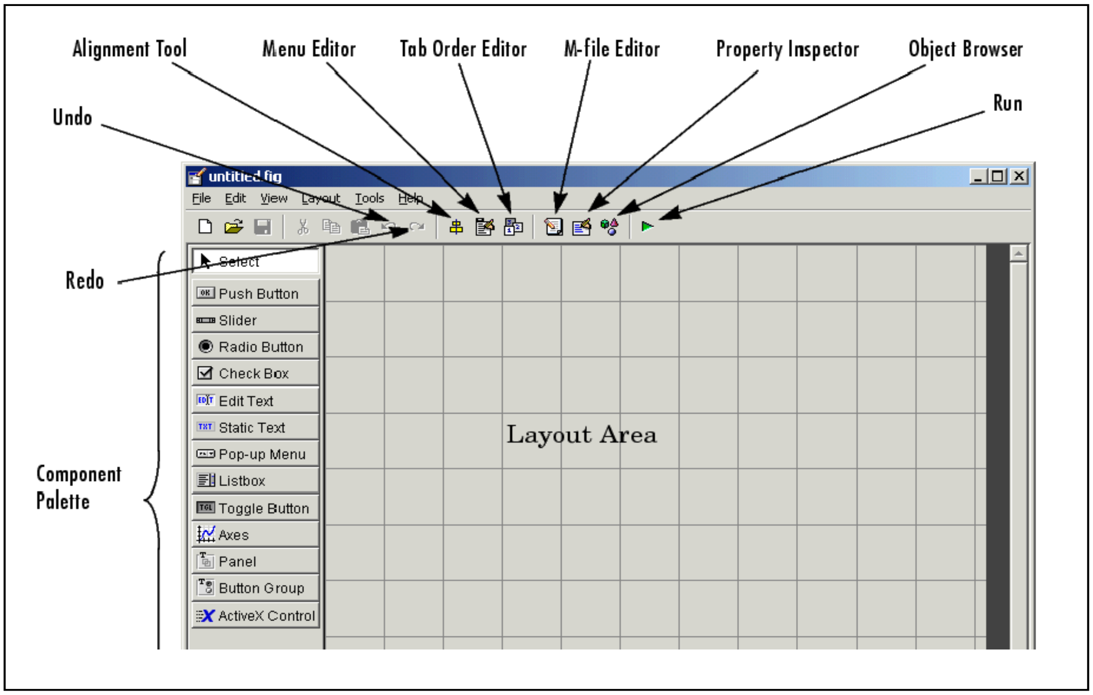


Figure 3.4 GUI Page

3.3.2.1 Component Palette

Table 3.1 : Name and Function of Component Palette

Name of Component	Function
Push Button	Push buttons will generate an action when clicked. For example, an OK button might close a dialog box and apply settings. By clicking the push button, it will appears depressed and by releasing the mouse, the button appears raised and its callback executes.
Toggle Button	Toggle buttons will generate an action and indicate whether they are turned on or off. When toggle button is click, it appears depressed, showing that it is on but when mouse button is release, the toggle button's callback will executes. However, unlike a push button, the toggle button remains depressed until the toggle button click at the second time.
Radio Button	Radio buttons are similar to check boxes, but are typically mutually exclusive within a group of

	related radio buttons. That's mean, only one button can be select at any given time. To activate a radio button, click the mouse button on the object. The display indicates the state of the button.
Check Box	Check boxes generate an action when checked and indicate their state as checked or not checked. Check boxes are useful when providing the user with a number of independent choices that set a mode, for example, displaying a toolbar or generating callback function prototypes.
Edit Text	Edit text controls are fields that enable users to enter or modify text strings. Edit text is use when the input is a text. The String property contains the text entered by the user. The callback executes when Enter is press for a single-line edit text, Ctl+Enter for a multi-line edit text, or the focus moves away.
Static Text	Static text controls display lines of text. Static text is typically used to label other controls, provide directions to the user, or indicate values associated with a slider. Static text cannot change interactively and there is no way to invoke the callback routine associated with it.
Slider	Sliders accept numeric input within a specific range by enabling the user to move a sliding bar, which is called a slider or thumb. Users move the slider by pressing the mouse button and dragging the slider, by clicking in the trough, or by clicking an arrow. The location of the slider indicates a percentage of the specified range.
List Box	List boxes display a list of items and enable users to select one or more items.
Pop-Up Menu	Pop-up menus open to display a list of choices when users click the arrow.
Axes	Axes will able GUI to display graphics (e.g., graphs and images). Like all graphics objects, axes have properties that can be set to control many aspects of its behavior and appearance.
Panel	Panels group GUI components. Panels can make a user interface easier to understand by visually grouping related controls. A panel can have a title and various borders. Panel children can be panels and button groups as well as axes and user interface controls. The position of each component within a panel is interpreted relative to the panel.

Button Group	Button groups are like panels but can be used to manage exclusive selection behavior for radio buttons and toggle buttons. For radio buttons and toggle buttons that are managed by a button group, the code must be include to control them in the button group's Selection ChangeFcn callback function, not in the individual uicontrol Callback functions. A button group overwrites the Callback properties of radio buttons and toggle buttons that it manages.
ActiveX Component	ActiveX components enable you to display ActiveX controls in your GUI. See Adding an ActiveX Control to a GUI for more information and an example. Only figures can have child ActiveX components. Panels and button groups cannot. ActiveX components are available only on the Microsoft Windows platform.

3.3.3 Property Inspector

To set the properties of each GUI component, the *Property Inspector* from the *View* menu is chosen to display the Property Inspector dialog box. When the component in the Layout Editor is chosen, the Property Inspector will displays that component's properties. If no component is selected, the Property Inspector displays the properties of the GUI figure.

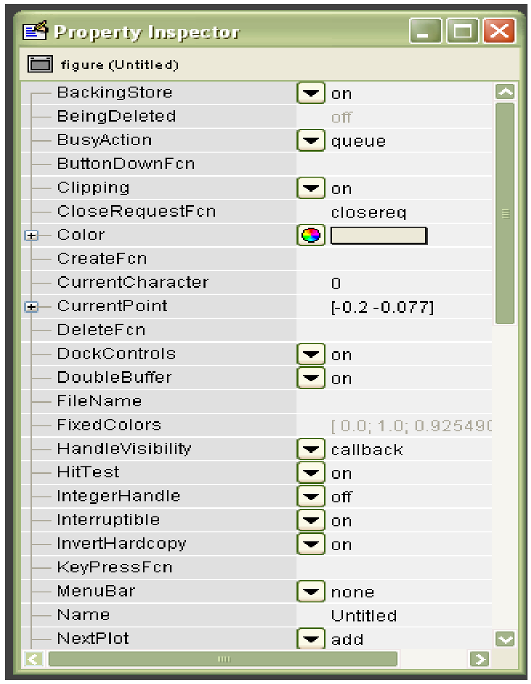


Figure 3.5 Property Inspector

The Property Inspector is used to give each component a name and to set the characteristics of each component such as color, name, font size, font name, tag and others.

3.4 Program the GUI

GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for all the GUI callbacks -- the commands that are executed when a user clicks a GUI component.

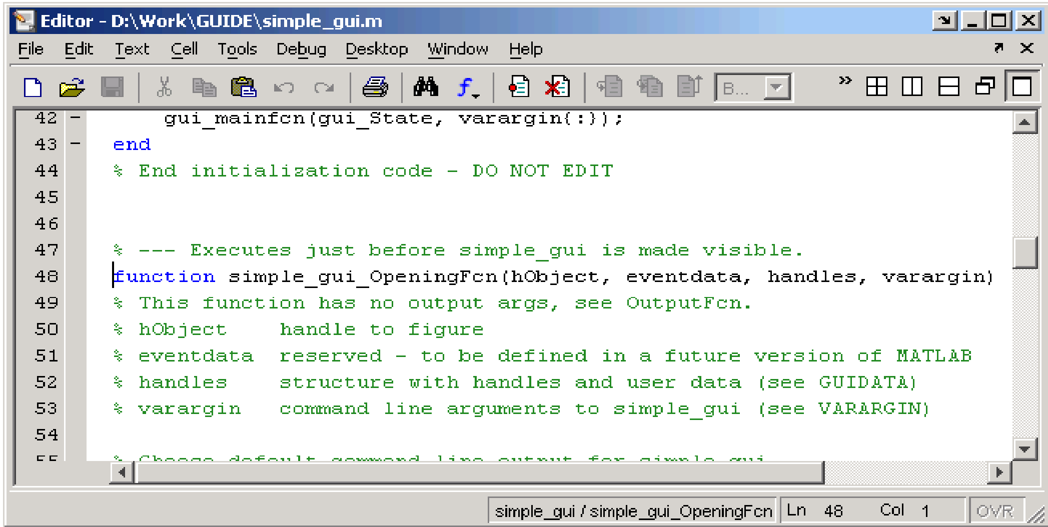


Figure 3.6 M.file Example

Figure 3.6 was the example of M.file that is generated when the complete layout has been designed and saved. By using M.file, the behavior of the GUI can be programmed by several code or function. This code will be programmed to give responsed for component palette that has been designed in the Layout area.

The programming M.file was the most difficult part in developing GUI because it need an extra reading on the coding before GUI can be performed.

3.5 Problem Simulation

For simulation of Power System Analysis using MATLAB 7.1, it will be discuss on:

i) Optimal Dispatch of Power Generation Software Package Using MATLAB
ii) Toolbox for Power System Fault Analysis using MATLAB
iii) Power Flow Analysis Software using MATLAB
iv) Steady-state and Transient Stability Analysis with MATLAB

After laying out the GUI component and set the property, the GUI will be look like in Figure 3.7.

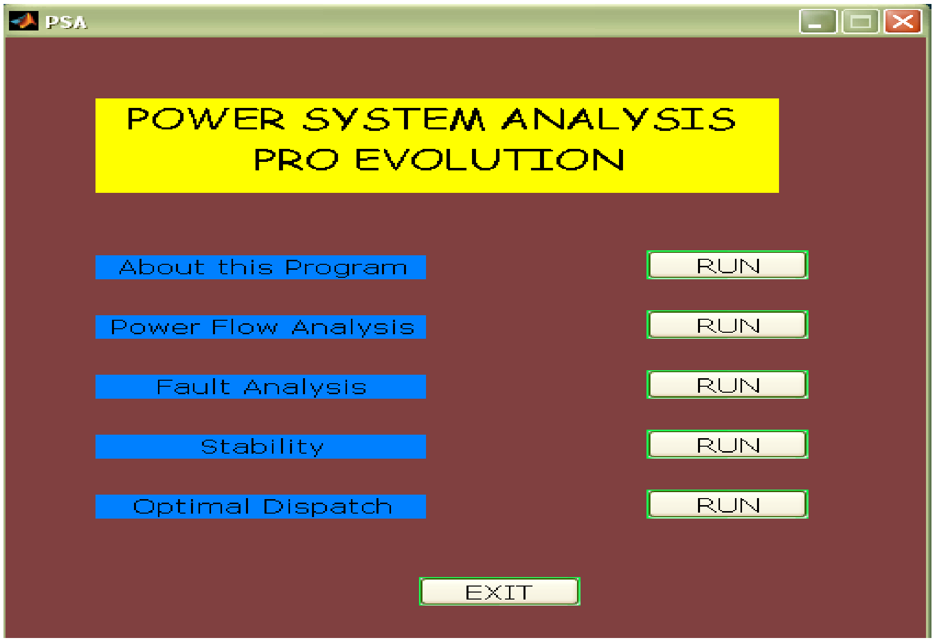


Figure 3.7 Layout for Compilation Project using MATLAB.

In the Figure 3.7, there are only 2 component that be used. There are:

a) *Static text* that use to display Power System Analysis Pro Evolution, About This Program, Power Flow Analysis, Fault Analysis, Stability Analysis and Optimal Dispatch.

b) *Push button* that use to *RUN* and *EXIT* the program. By clicking on the *RUN* button, this program will execute depend on type of analysis that are be chosen. If the *EXIT* button is click, this window will close and bring back to MATLAB main window.

The example of the M.file that is generated from the GUI window s shown in Figure 3.8.

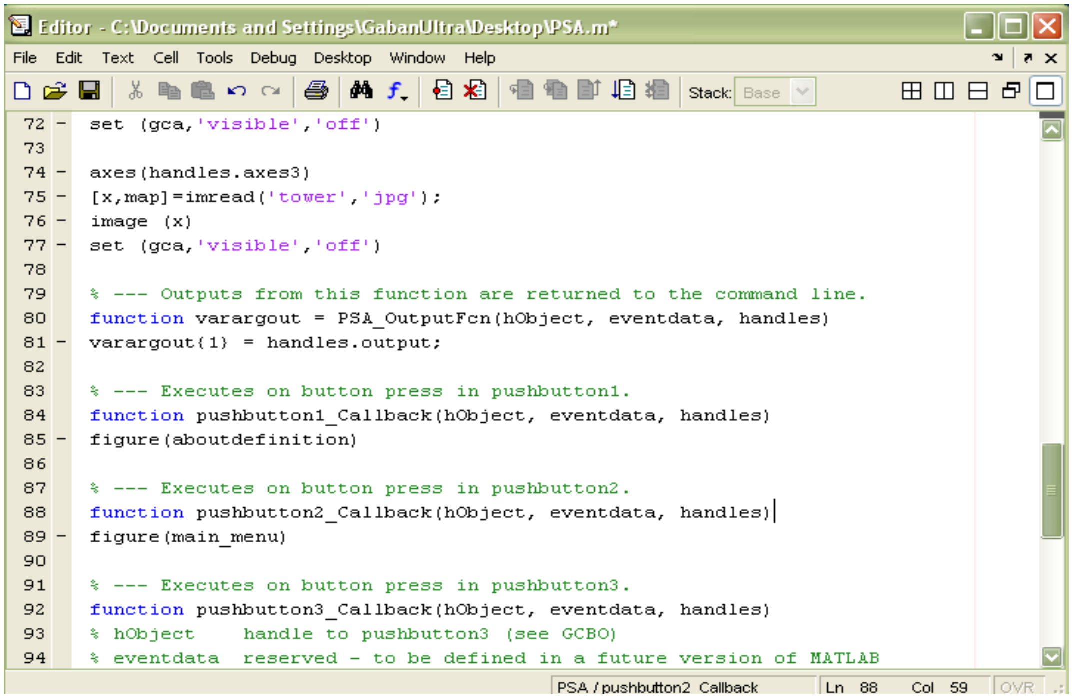


Figure 3.8 M.file Program

Figure 3.8 is shows the example of M.file that be generated when the completed layout like in Figure 3.7 have been saved. Using the M.file, the behavior of the GUI can be programmed by several codes. This code will controls on how the GUI respond to events such as push button, slider, menu item selection, or the creation and deletion of component. This programming takes the form of set of functions, called callbacks, for each component and for the GUI itself.

All of the result about this project will be described at Chapter 4.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 User Guide On How To Start Power System Analysis Pro Evolution Toolbox

Power System Analysis Educational Toolbox will provide four (4) module which are, Power Flow Analysis, Fault Analysis, Stability Analysis and lastly, Optimal Dispatch of Power Generation. Each of analysis is built with examples and tutorial to help user on how to use this Pro Evolution toolbox.

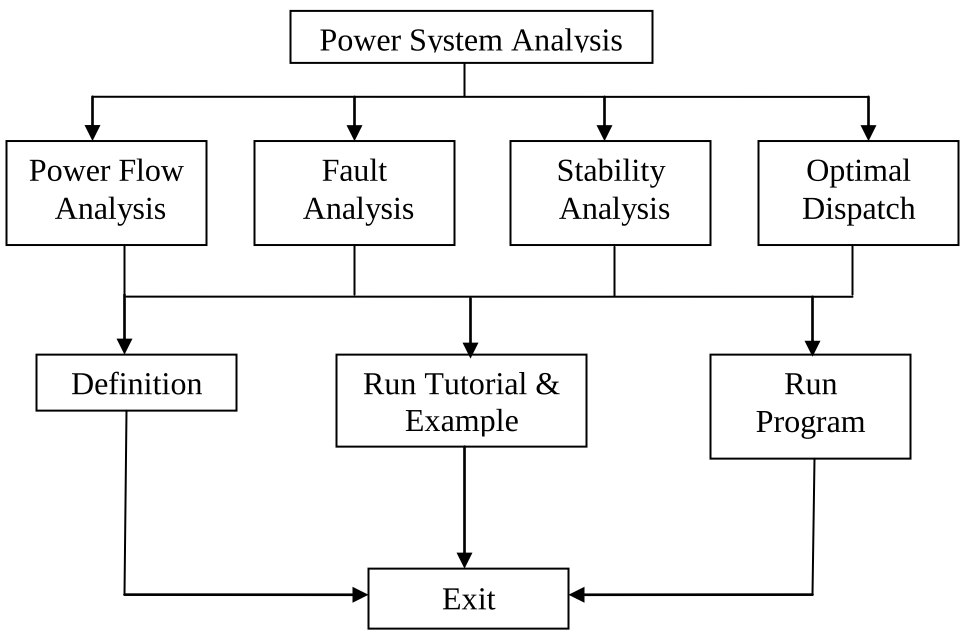


Figure 4.1 Flow Chart of Power System Analysis Educational Toolbox

To use the software, the user must install MATLAB 7.1 software in computer. Then, user can click on MATLAB 7.1 software icon at the desktop or the user can find it in *Start* program. The window will appear like this:

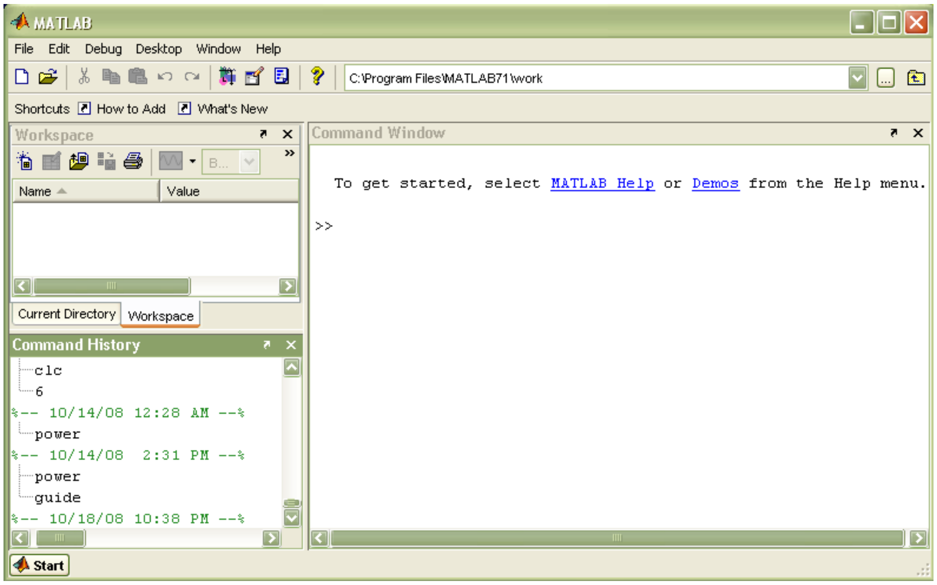


Figure 4.2 Main Window of MATLAB 7.1 Software

All the work that is done by the author has been compiled in a folder named as Power. This folder had be located at C:\Program Files\MATLAB71\Power. The user must ensure that all the folders is located in drive C in MATLAB 7.1 file folder. If not, the user may have problems to run this program. Error will appear if it saved in different drive and location.

Then, the “Add Path” icon is chosen in order to add the project file in Matlab folder. The step is shown in Figure 4.3.

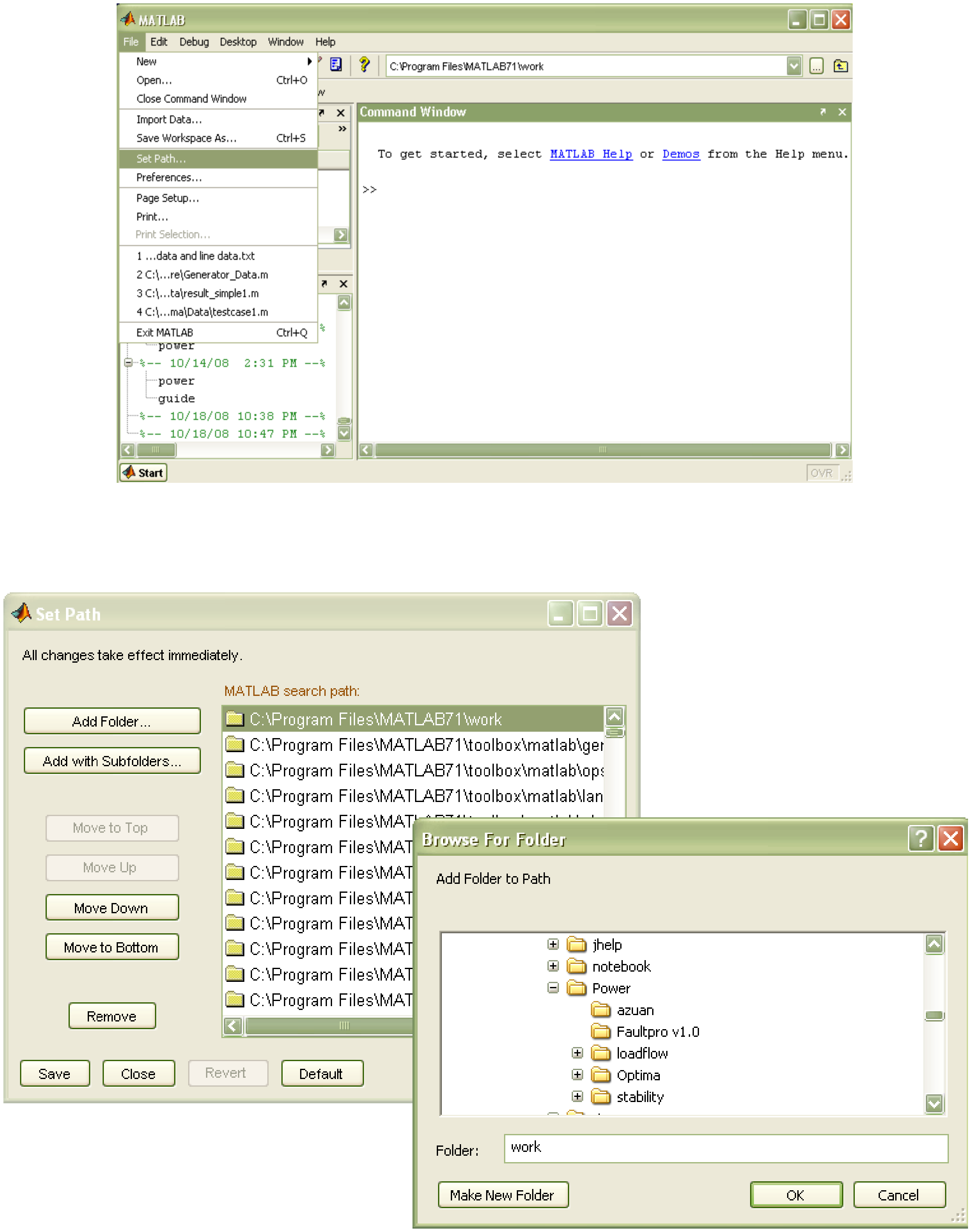


Figure 4.3 Add Path

The user must add all files in the Power folder and save before close the window.

4.1.1 Power System Analysis Educational Toolbox

After the user had done the "Add Path" part, the user now can use this software. On main window of the software, the user may type the word 'Power' or 'power' or 'POWER'. This step will run the main window of this Educational Toolbox.

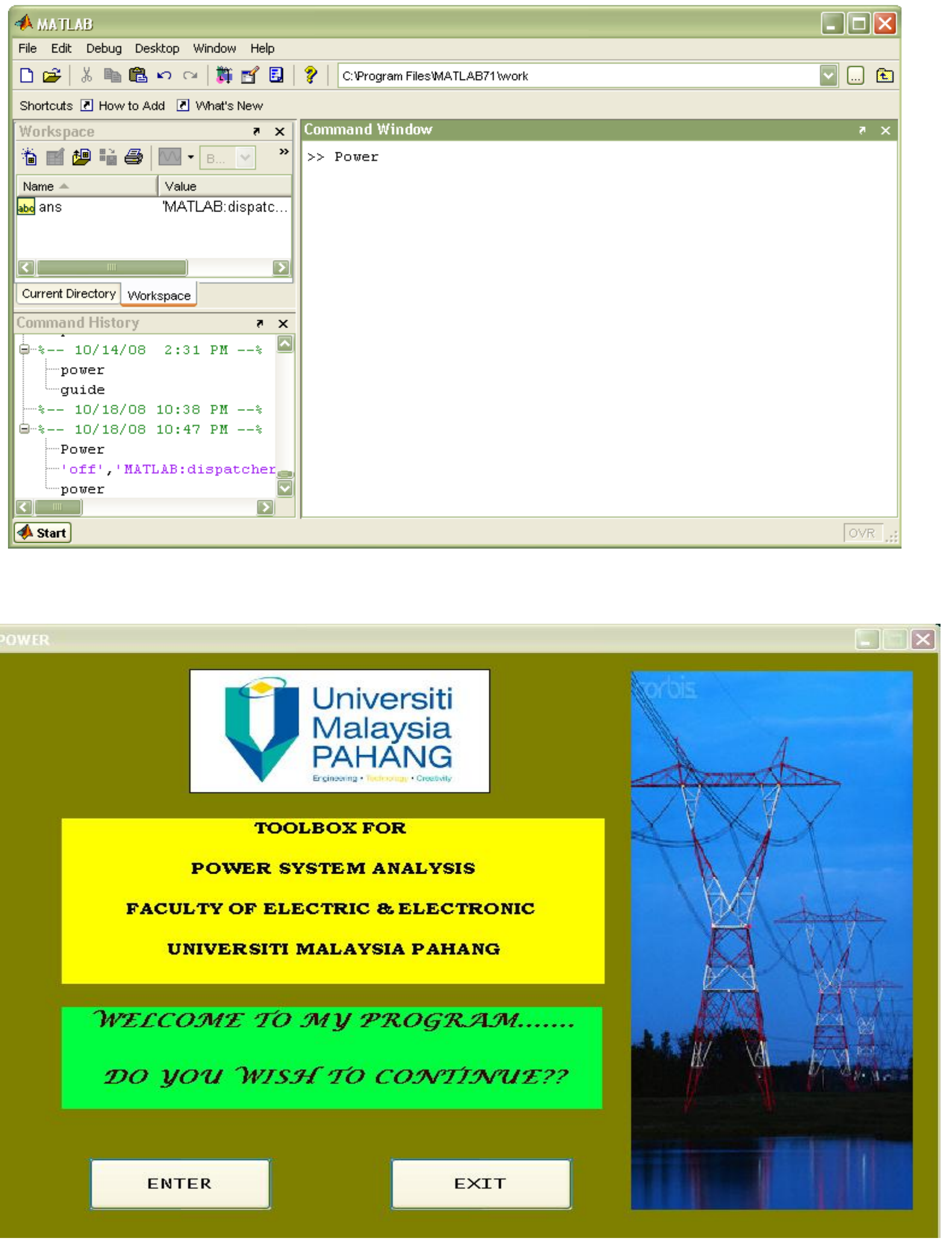


Figure 4.4 Main Page of Power System Analysis Educational Toolbox

On this main page, there are TWO (2) different pushbuttons. If the user clicks on *EXIT*, the window will close automatically. But, if the user click on *ENTER* button, Figure 4.5 will appear.

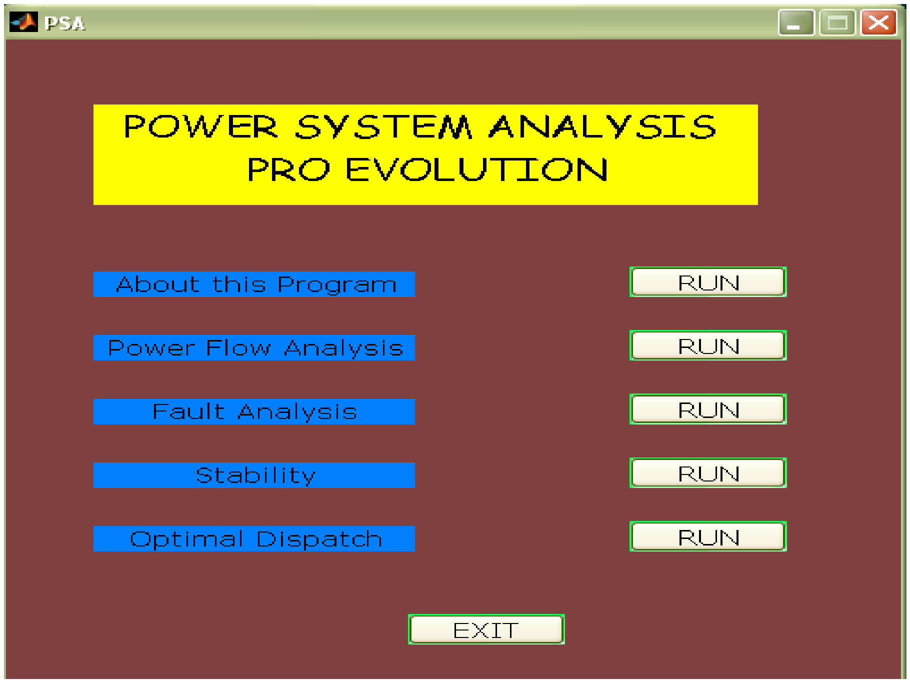


Figure 4.5 Power System Analysis Pro Evaluation

4.2 User Guide On How To Use Power System Analysis Educational Toolbox

The main page of this project has five (5) pushbuttons which is, each of the pushbutton named as 'RUN'. These pushbuttons will open each new window depend on what you want to do. The window is shown as Figure 4.5

4.2.1 Definition of this Project

The first pushbutton 'RUN' will open the new window named as About Definition. This can be seen as in Figure 4.6.

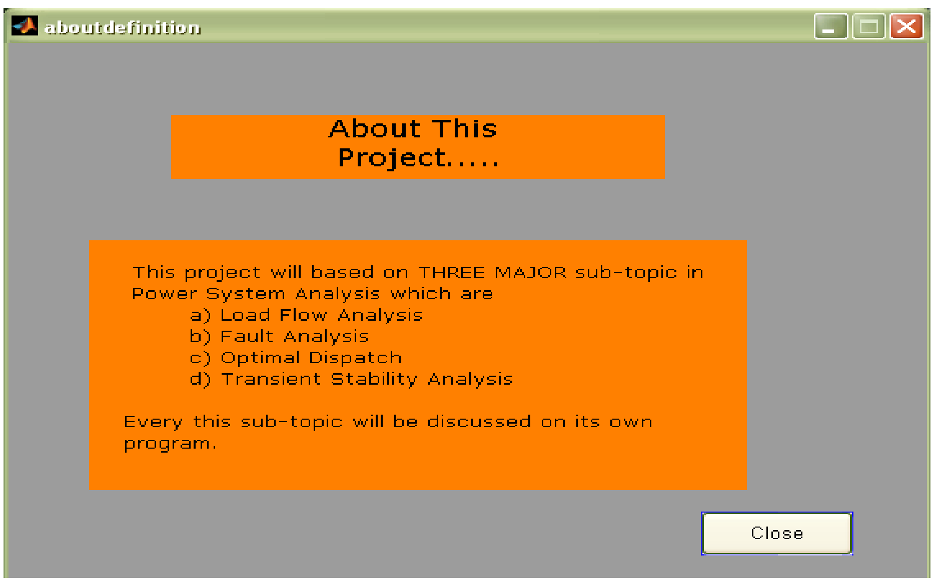


Figure 4.6 About Definition

4.2.2 Power Flow Analysis

The second pushbutton will display the Power Flow Analysis program as shown in Figure 4.7.

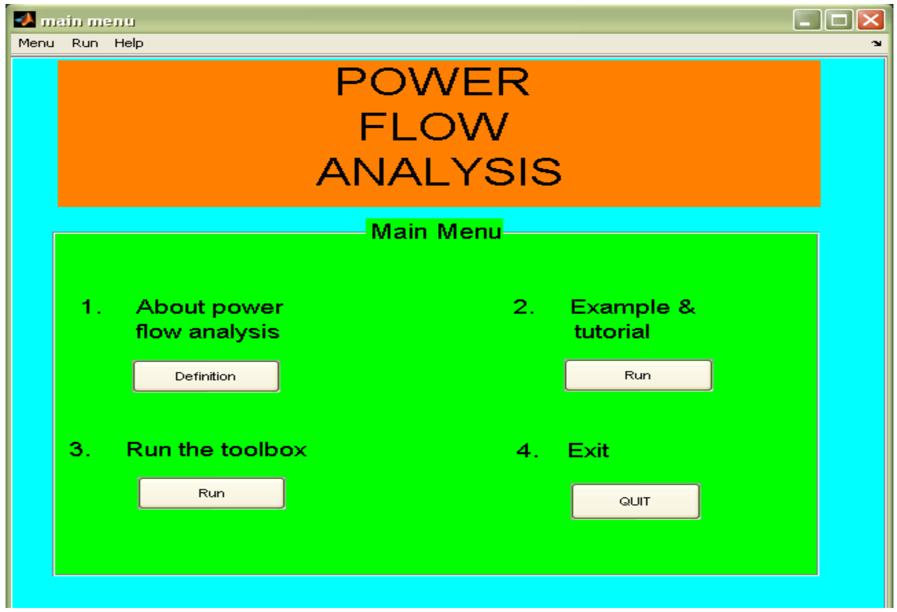


Figure 4.7 Main Menu of Power Flow Analysis

4.2.3 Fault Analysis

The third pushbutton will described on Fault Analysis as shown in Figure 4.8.

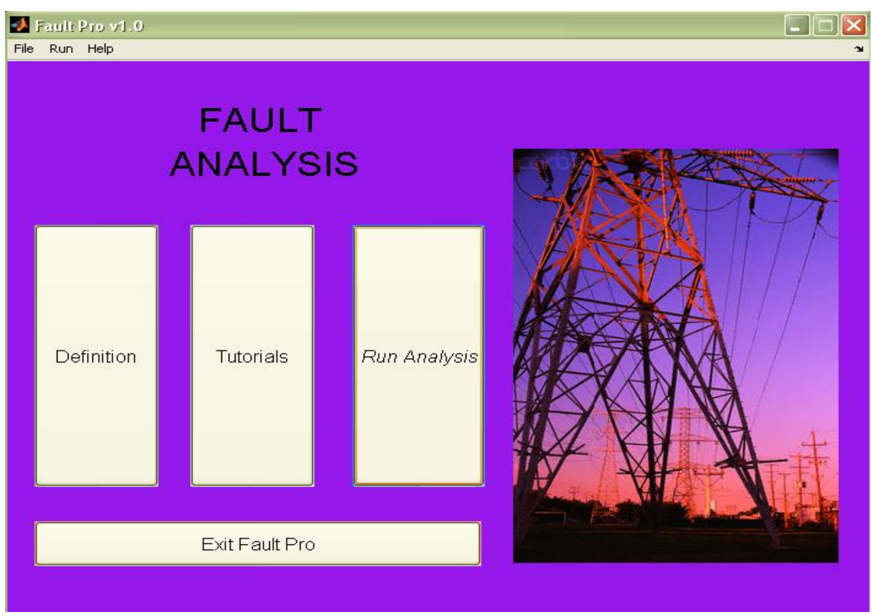


Figure 4.8 Main Page of Fault Analysis

4.2.4 Stability Analysis

The fourth pushbutton will display Stability Analysis as shown in Figure 4.9.

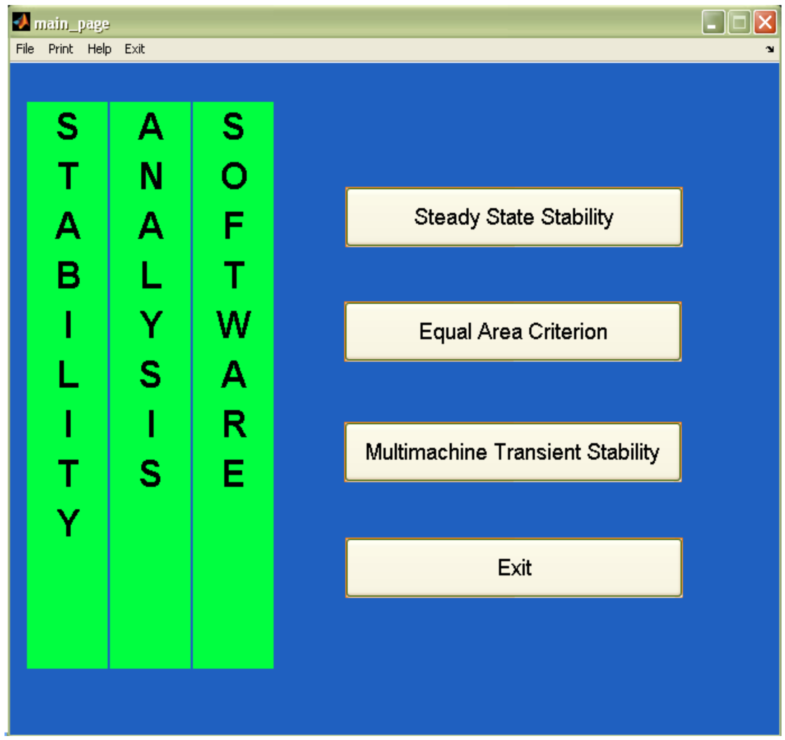


Figure 4.9 Stability Analysis

4.2.5 Optimal Dispatch of Power Generation

The fifth pushbutton will display about Optimal Dispatch when user clicks on it. These can be seen like Figure 4.10 below.

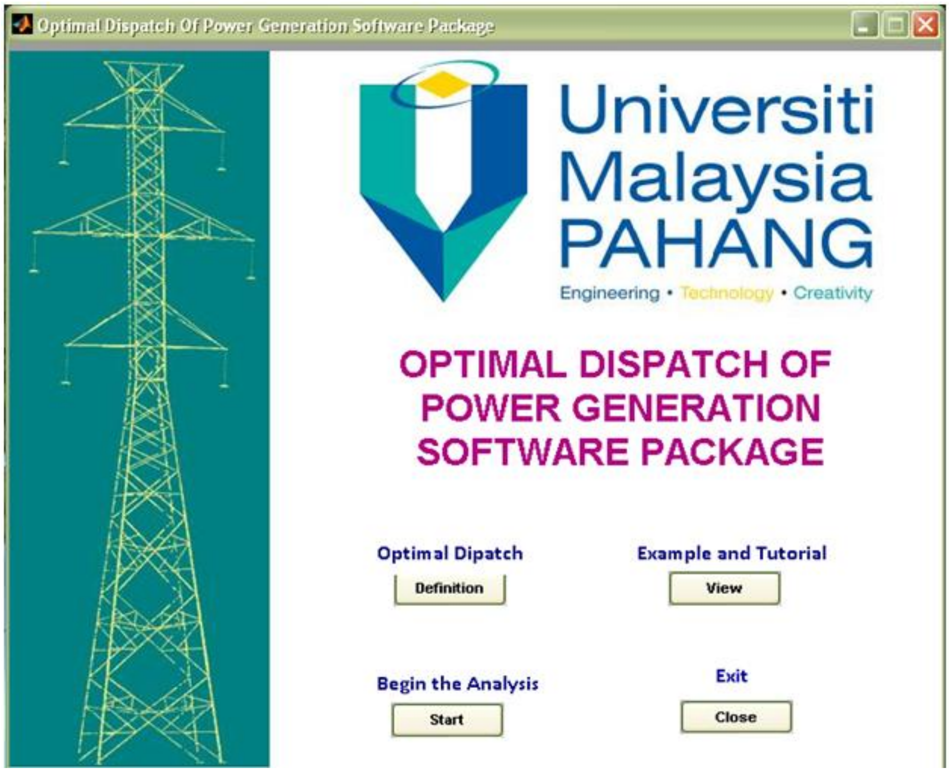


Figure 4.10 Optimal Dispatch of Power Generation's Main Page

4.3 User Guide On How To Use Power System Analysis Educational Toolbox (Power Flow Analysis)

There are four (4) pushbuttons. Each pushbutton will link to About Power Flow Analysis, Example & Tutorial, Run the Toolbox (own exercise) and lastly is Exit pushbutton (Refer to Figure 4.7: Main Menu of Power Flow Analysis).

4.3.1 About the definition

If the user clicks on Definition pushbutton, Figure 4.11 will appear. It will tell the user about the definition of Power Flow.

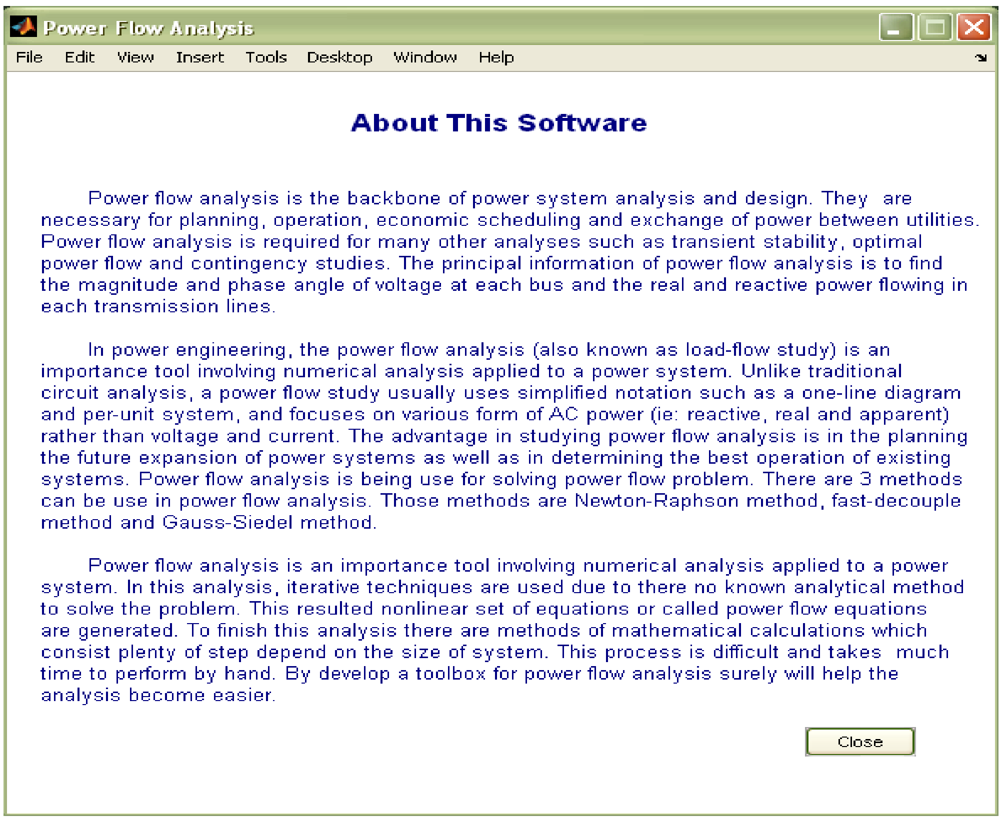
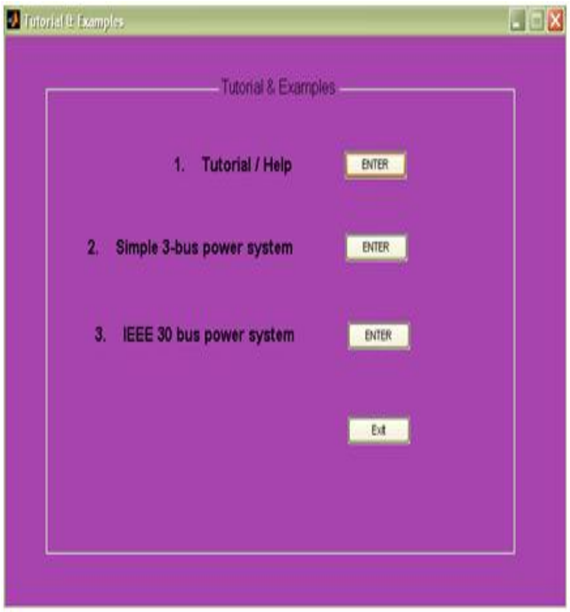


Figure 4.11 Definition of Power Flow Analysis

4.3.2 Example and Tutorial

On this sub-topic, the user will be introduces on Tutorial/Help, example of Simple three (3) bus and IEEE 30-bus Power system. It can be shown as Figure 4.12 below.

On Tutorial/Help section, it will help the user to use this software like Figure 4.13 below. The figure will teach the user on how to fill in the bus data and line data. The user can click on View button to see more clearly.



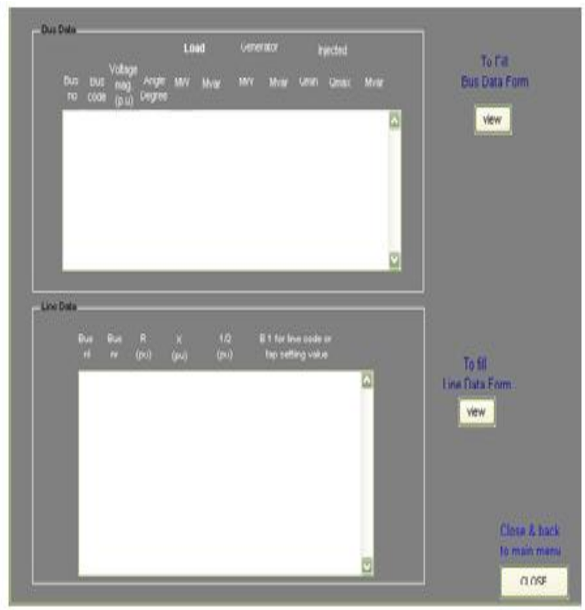


Figure 4.12 Tutorial & Example

Figure 4.13 Tutorial/Help

The *ENTER* button of Simple 3-bus Power System will describe about a simple question on 3-bus. When user click the button, next window will appear and data that need to be key in, was already enter by the author. It can be seen like Figure 4.14 below. The user can proceed with click on Proceed button and next window will appear such Figure 4.15 below. To see the result, the user must click on Power Mismatch button on each type of solution to get the result. The line flow and losses will appear on main

Matlab 7.1 menu after the user click on Total Load Flow button as shown in Figure 4.16.

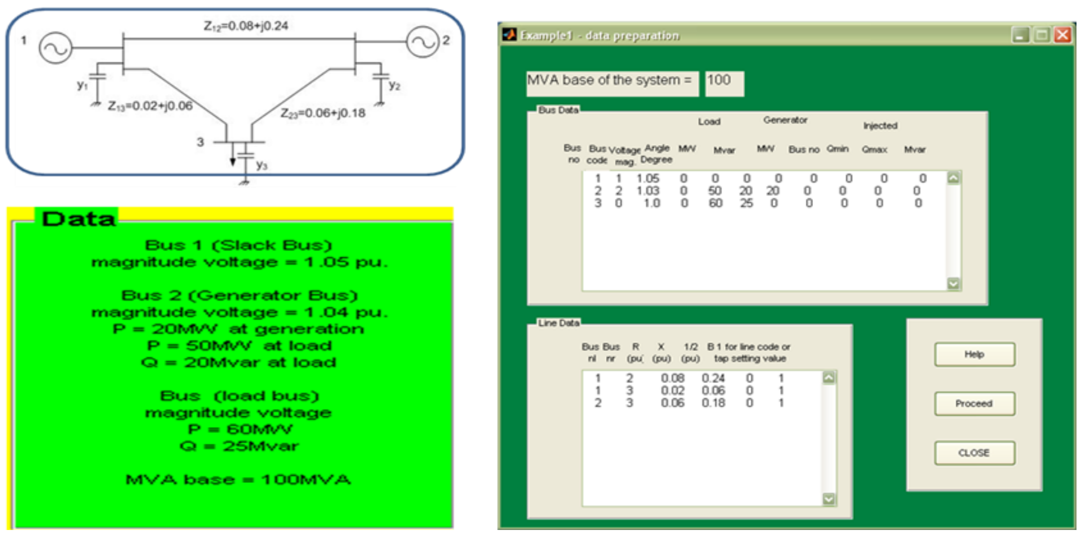


Figure 4.14 Simple 3-bus power System

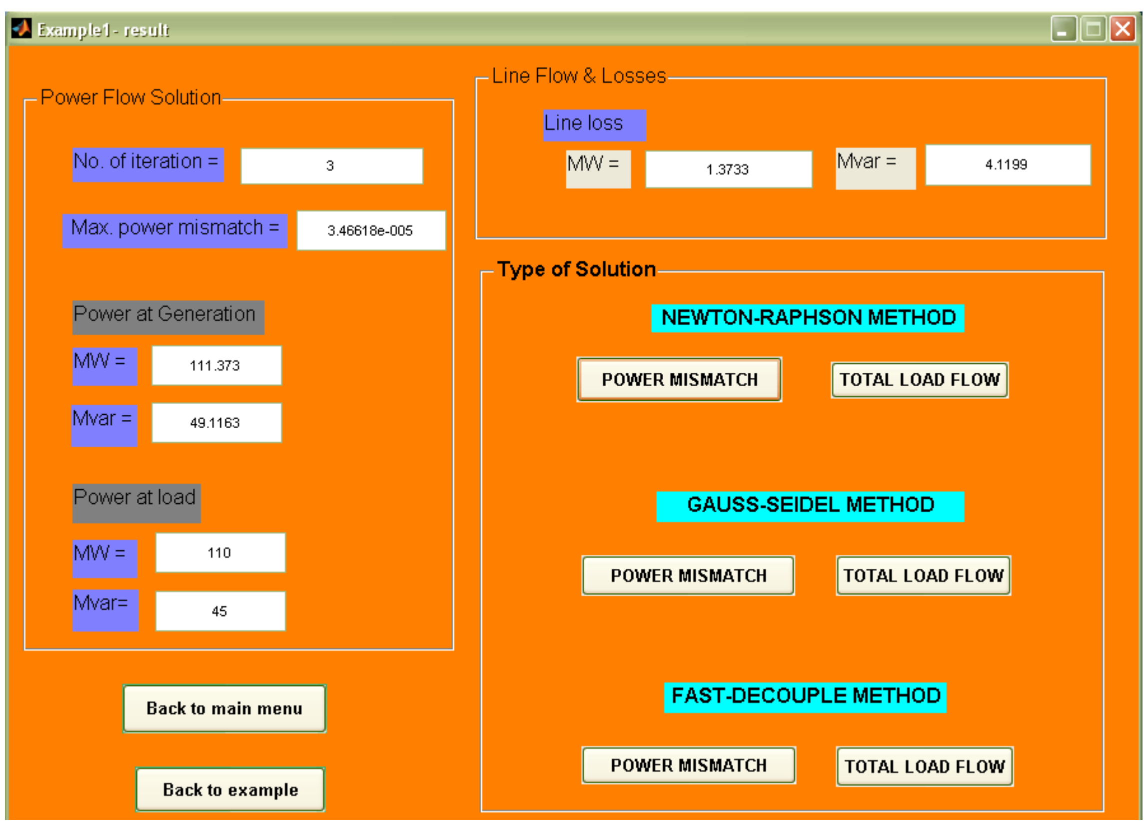


Figure 4.15 Result of Simple 3-bus power System

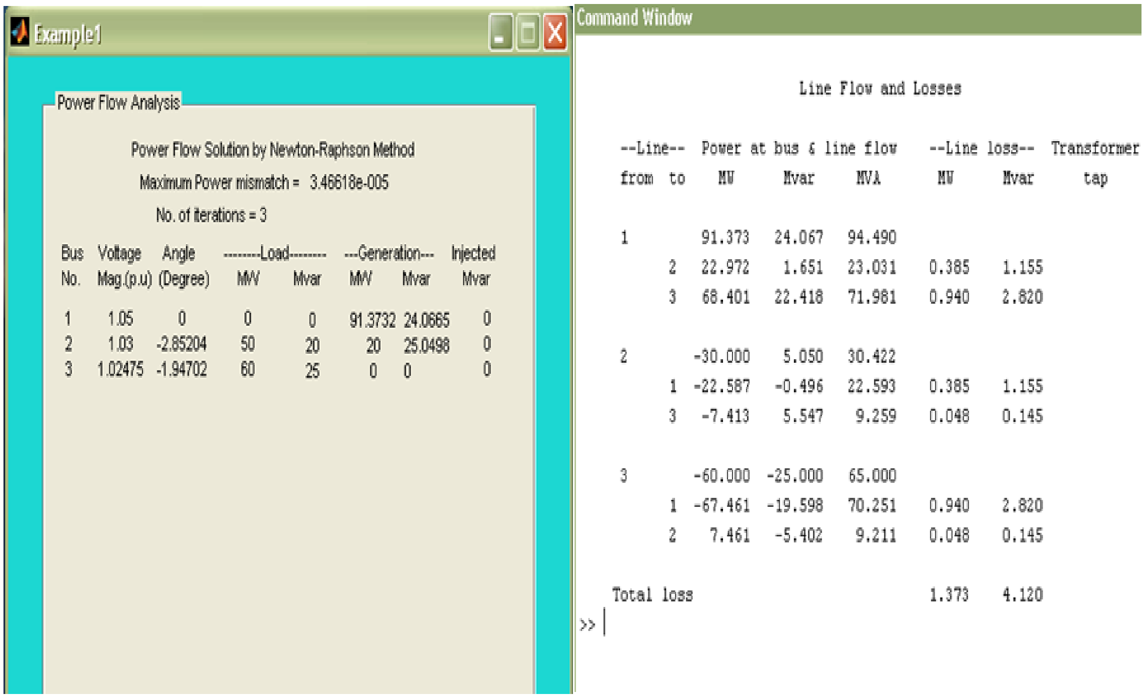


Figure 4.16 Example of Newton-Raphson method

For IEEE 30-bus Power System, the method on how to gets the result are quiet same with simple 3-bus. The difference is, the bus data and line must be changed depend on the question. The example of IEEE 30-bus power system shown in Figure 4.17 below.

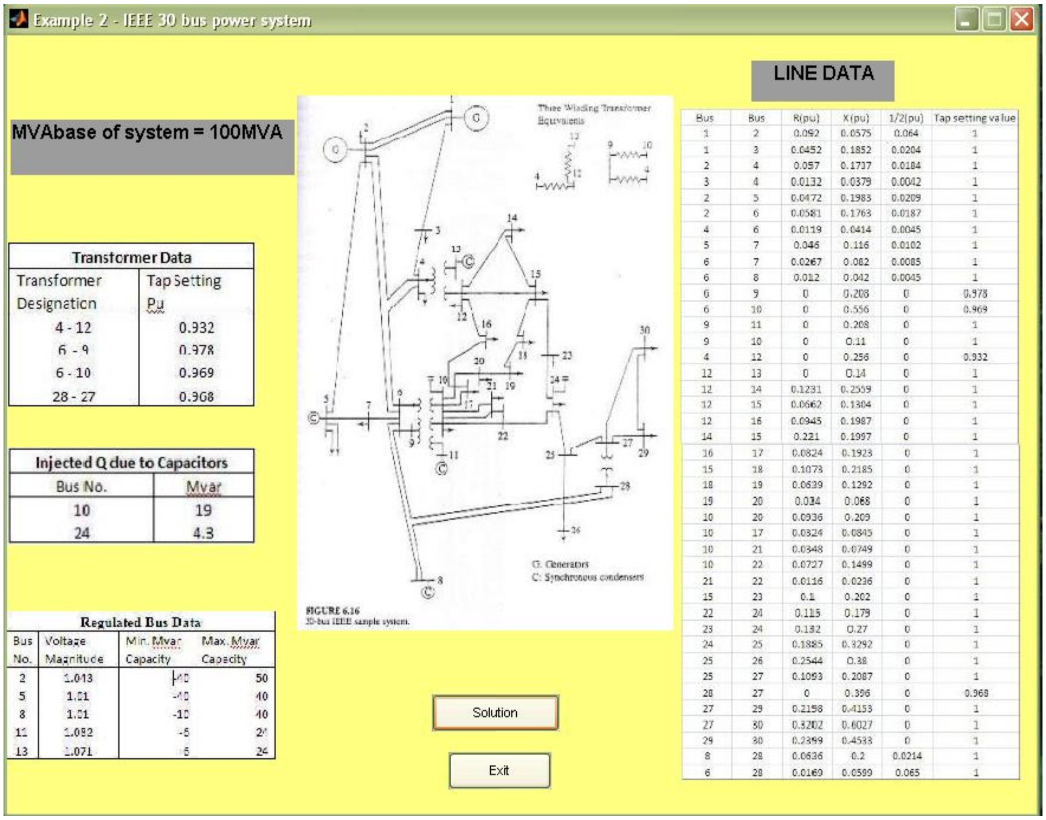


Figure 4.17 Example of IEEE 30-bus system

4.3.3 Run the Toolbox

When a new simulation is needed, the user can click on Run the Toolbox pushbutton. The new window will appear as shown in Figure 4.18 and all the data must be key in like in previous example. After the user had complete enter the input data, the user can click on the Proceed pushbutton and choose type of solution like Figure 4.19. There are 3 methods of solution discussed in this toolbox that are Newton-Raphson method, Gauss-Seidel Method aand Fast Decoupled method.

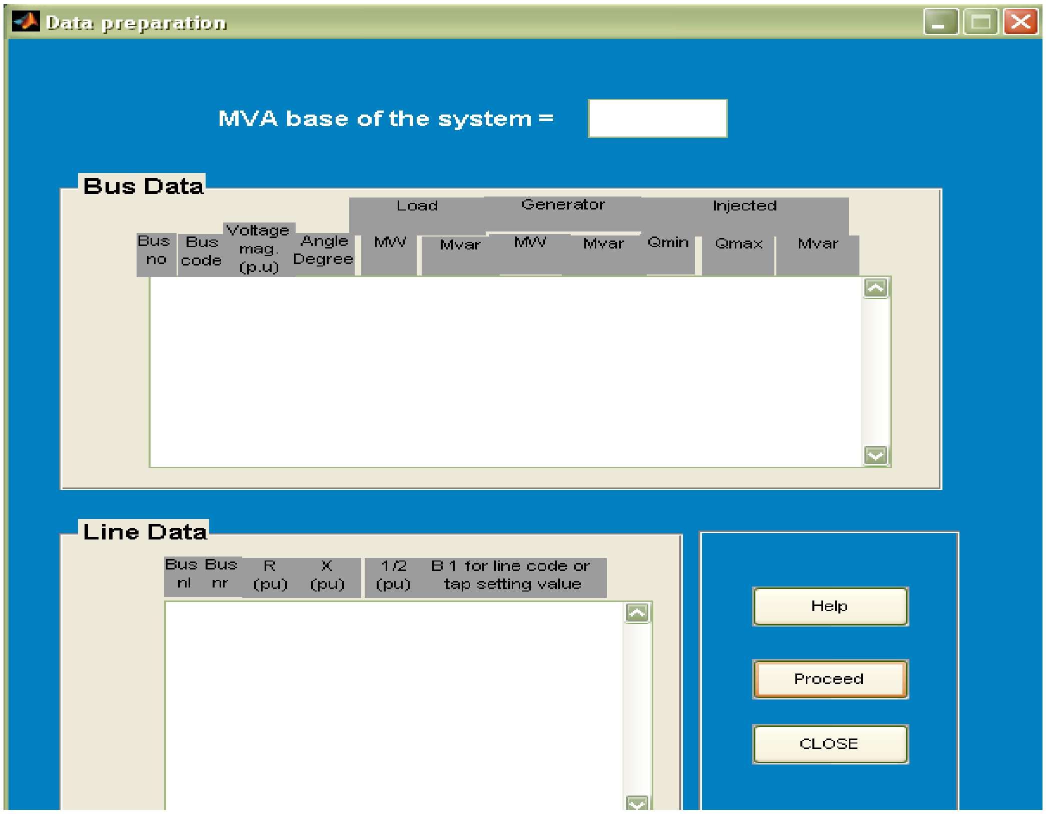


Figure 4.18 Run Toolbox Layout

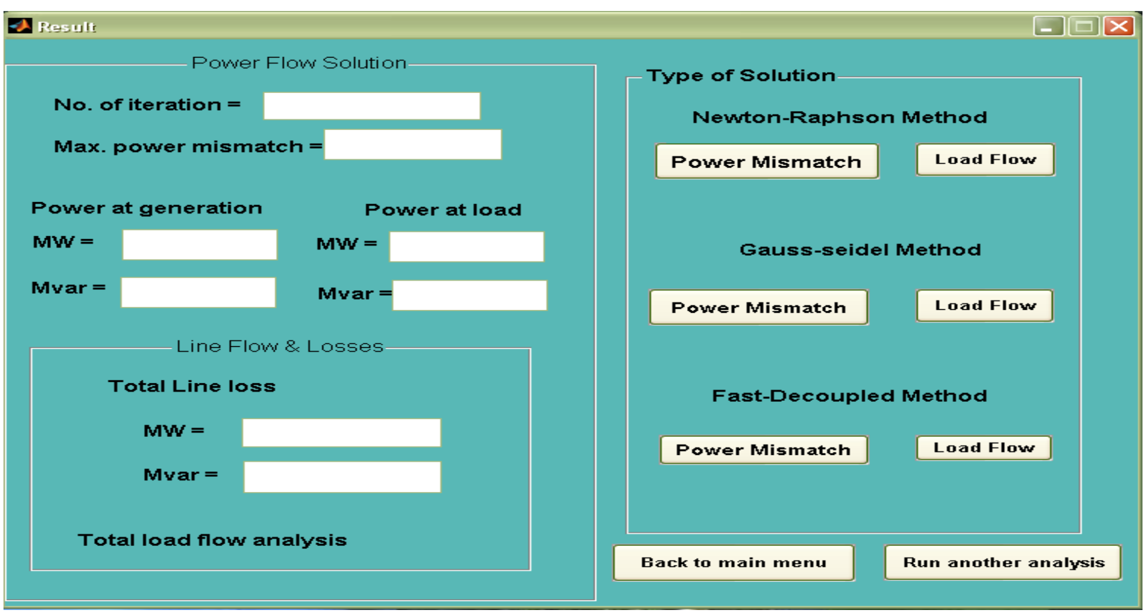


Figure 4.19 Types of Solution Layout

4.4 User Guide On How To Use Power System Analysis Educational Toolbox (Fault Analysis)

The main page of Fault Analysis is like Figure 4.8. On this topic, there are four pushbuttons has been created. First pushbutton will display the Definition, second pushbutton is Tutorial, third pushbutton is Run Analysis and the last pushbutton is Exit. The Exit button will close the main page of Fault Analysis.

4.4.1 About Fault Analysis

If the user clicks on Definition button, Figure 4.20 will appear.

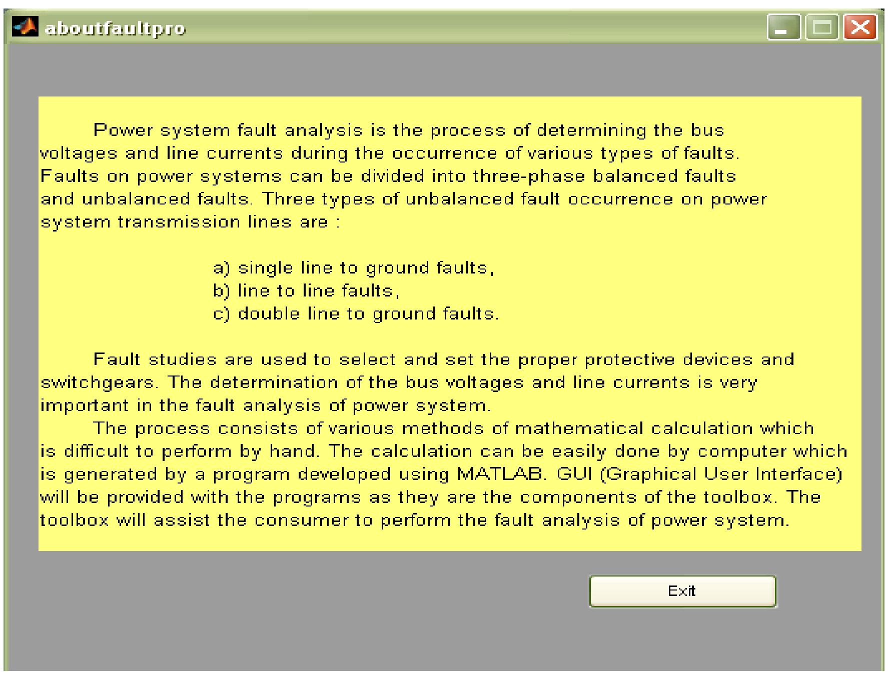


Figure 4.20 Definition Layout

4.4.2 Tutorial of Fault Analysis

If the user clicks on Tutorial button, Figure 4.21 will appear.

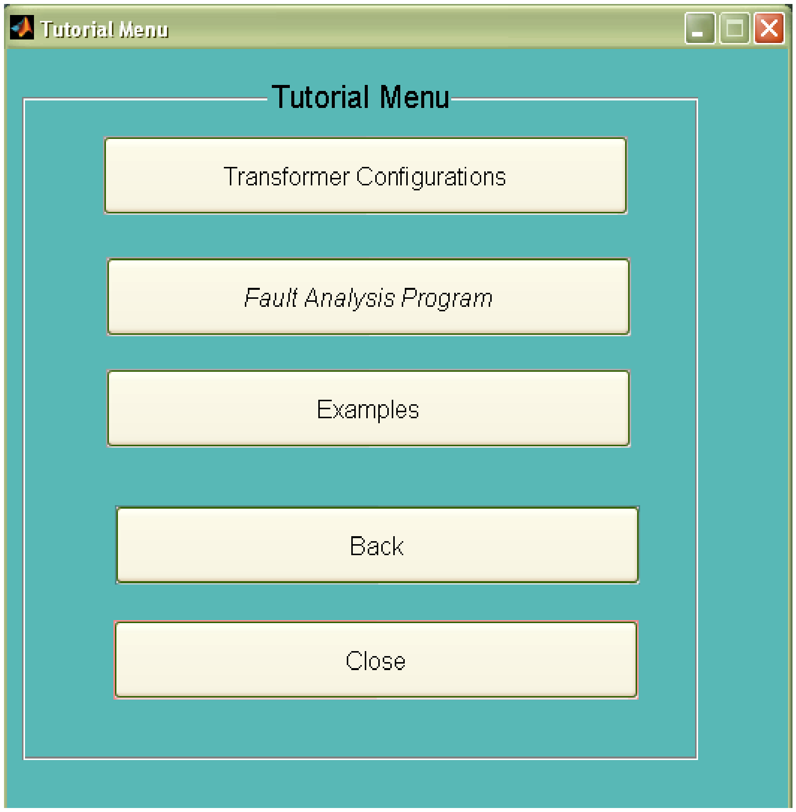


Figure 4.21 Tutorial Menu

In this page, five pushbuttons have been created. First pushbutton will discuss on *transformer configuration* like Figure 4.22 below. Such connection and configuration will produce different impedance calculation in the transmission line. For example, if the user clicks on wye-ground-delta, the new window will appear to guide the user to identify and differentiate each types of transformer connection and configuration. This is very important in order to identify zero sequence impedance of a system.

Second pushbutton is *Fault Analysis Program*. This pushbutton is chosen to prompt a new window named as Fault Tutor as shown in Figure 4.23.

Third pushbutton will discuss on *Example* of the question as shown in Figure 4.24. Three examples are provided which are, 3-bus, 11-bus and 32-bus system. For example of 3-bus system, the input data is shown in Figure 4.25. Then the user can choose type of fault that occured at the system. The faults are *Balanced Three Phase Fault*, *Single Line to Ground Fault*, *Double Line to Ground Fault*, and *Line to line Fault*. The result of *Balanced Three Phase Fault* for 3 bus system is shown in Figure 4.26. This window will appear at Matlab 7.1 main page. The user can also solve 11-bus system and 32-bus system which are provided in the toolbox.

The *Back* pushbutton will bring the user to main page of Fault Analysis and *Close* button, will close the window.

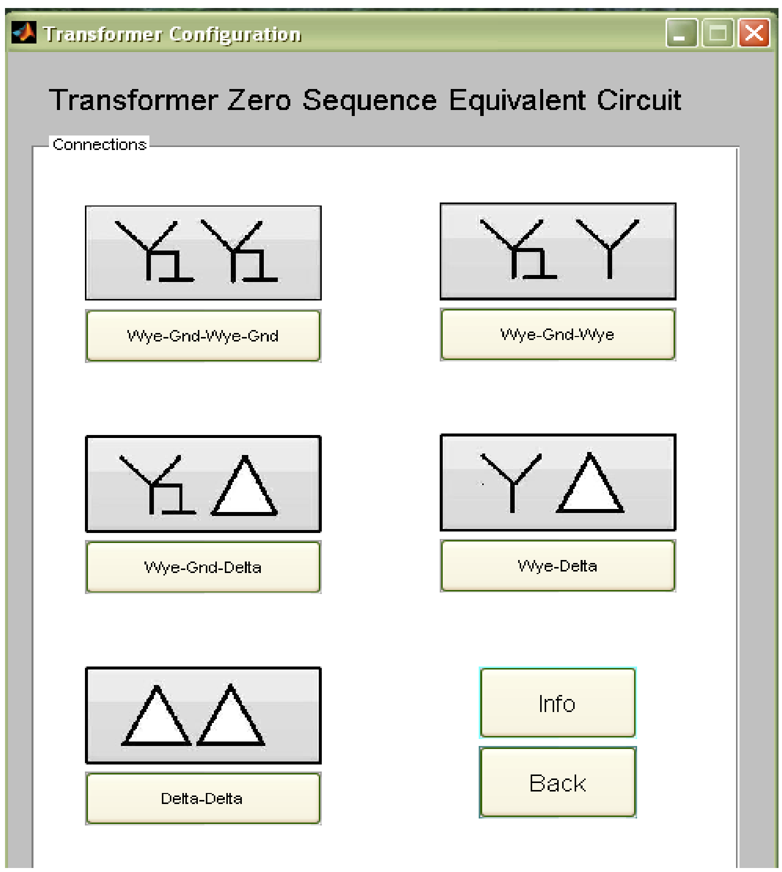


Figure 4.22 Transformer Configuration

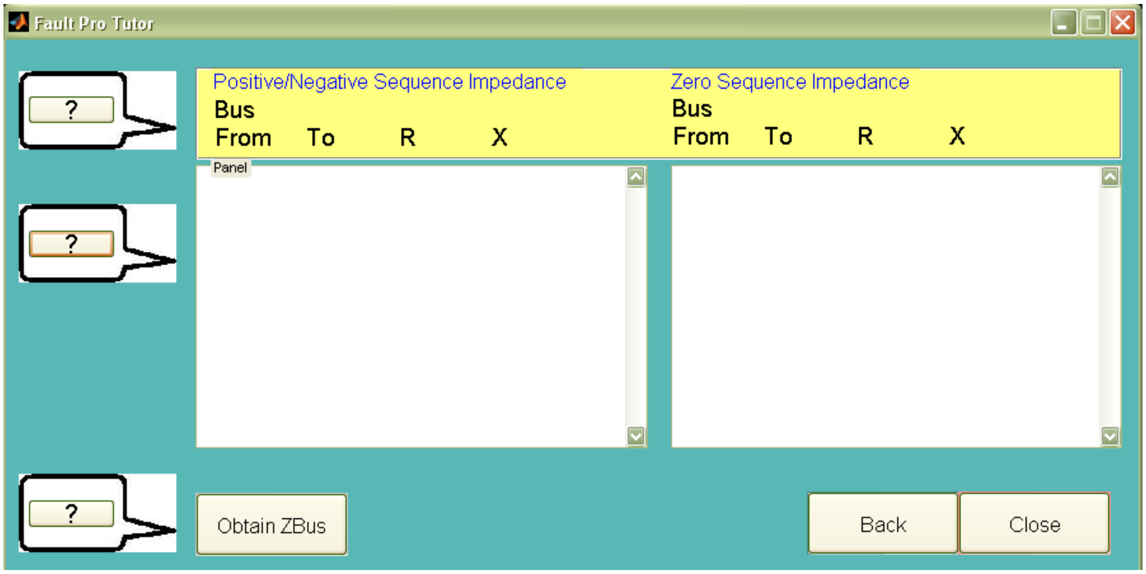


Figure 4.23 Fault Tutor Layout

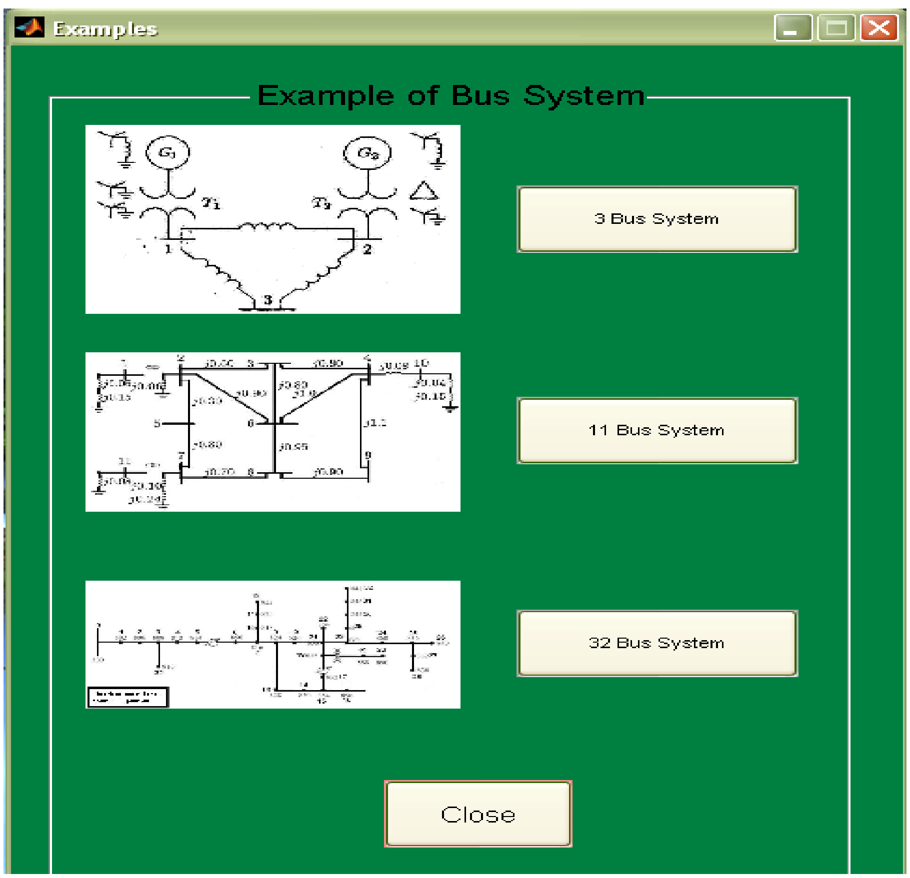


Figure 4.24 Example of Fault Analysis

Figure 4.25 Example of 3-bus System

```
-------------Three-phase balanced fault analysis--------------
Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 1.000000e-001
Balanced three-phase fault at bus No. 3
Total fault current =   3.1250 per unit

Bus Voltages during fault in per unit

     Bus      Voltage        Angle
     No.      Magnitude      degrees
      1        0.5938        0.0000
      2        0.6250        0.0000
      3        0.3125        0.0000

Line currents for fault at bus No.  3

     From        To     Current      Angle
     Bus         Bus    Magnitude    degrees
      G           1      1.6250     -90.0000
      1           3      1.8750     -90.0000
      G           2      1.5000     -90.0000
      2           1      0.2500     -90.0000
      2           3      1.2500     -90.0000
      3           F      3.1250     -90.0000

Faultpro v1.0 Balance Three Phase Fault
Universiti Malaysia Pahang
All Rights Reserved
-------------------------Analysis End-------------------------
```

Figure 4.26 Result of Balanced Three Phase Fault

4.4.3 Run Fault Analysis

If the user clicks on Run Analysis button, Figure 4.27 will appear.

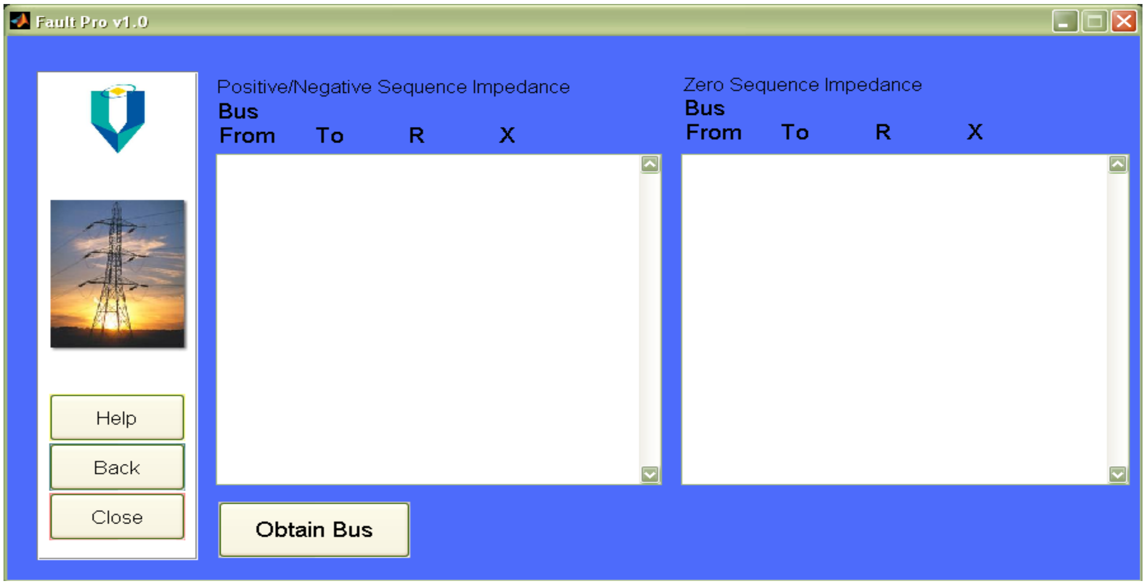


Figure 4.27 Run Analysis Layout

This layout can be used to obtain new simulation. All the input data must be entered as in previous example.

4.5 User Guide On How To Use Power System Analysis Educational Toolbox (Stability Analysis)

Other analysis that is provided in this educational toolbox is Stability Analysis. The main menu of this analysis can be seen in Figure 4.9. The stability will discuss steady state stability, Equal Area Criterion and Multimachine Transient Stability.

4.5.1 Steady-State Stability

In this analysis, first of all, the user needs to click the pushbutton that displays Steady-state Stability. A new window will display. All the empty parameter must be fill in with per unit value. After the parameter is entered, user can click on Plot button and the result of natural response of the rotor angle and natural response of the frequency will be shown. When user has finish in this section, the user may leave this window and back to main menu by clicking at return to selection when finish this task or click the exit button to exit whole stability analysis toolbox.

Figure 4.28 Example of Steady-State Stability

4.5.2 Equal Area Criterion

To use this simulation, the user as usual needs to click on equal area criterion pushbutton. A new window will appear as in Figure 4.29 below. There are two types of application that discussed here which are *Application of Equal Area Criterion due to sudden change in Power* and *Application of Equal Area Criterion due to Critically Cleared System*. The user also can back to main menu by clicking on Return to selection. If the user need guide, the user can clicks on Help button.

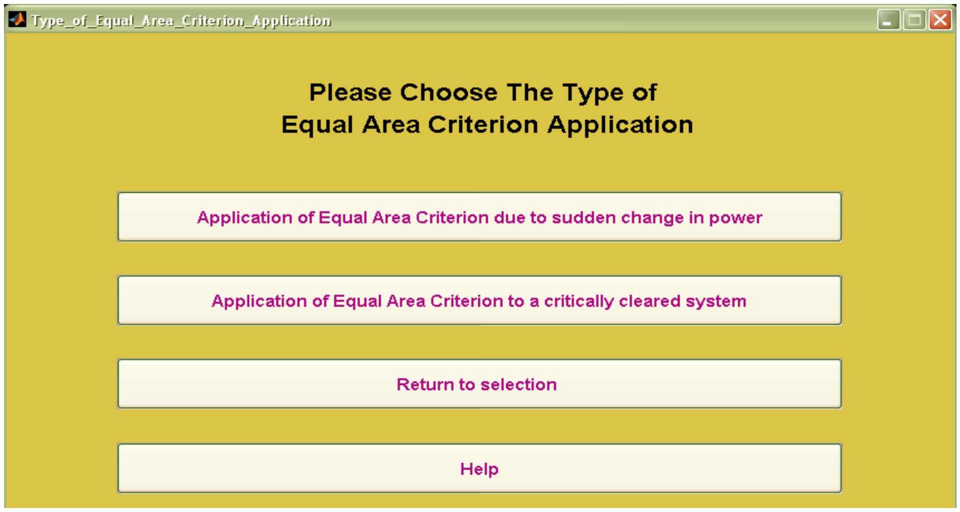


Figure 4.29 Equal Area Application

a) *Application of Equal Area Criterion due to sudden change in Power*

If this application is chosen, the new window will appear. The user needs to fill in the input parameter in per unit value like Figure 4.30 below. The Result pushbutton will display the answer of the analysis as shown in Figure 4.31.

b) *Application of Equal Area Criterion due to Critically Cleared System*

If this application is chosen, the new window will appear. The user needs to fill in the input parameter in per unit value like Figure 4.32 below. The Result pushbutton will display the answer of the analysis as shown in Figure 4.33.

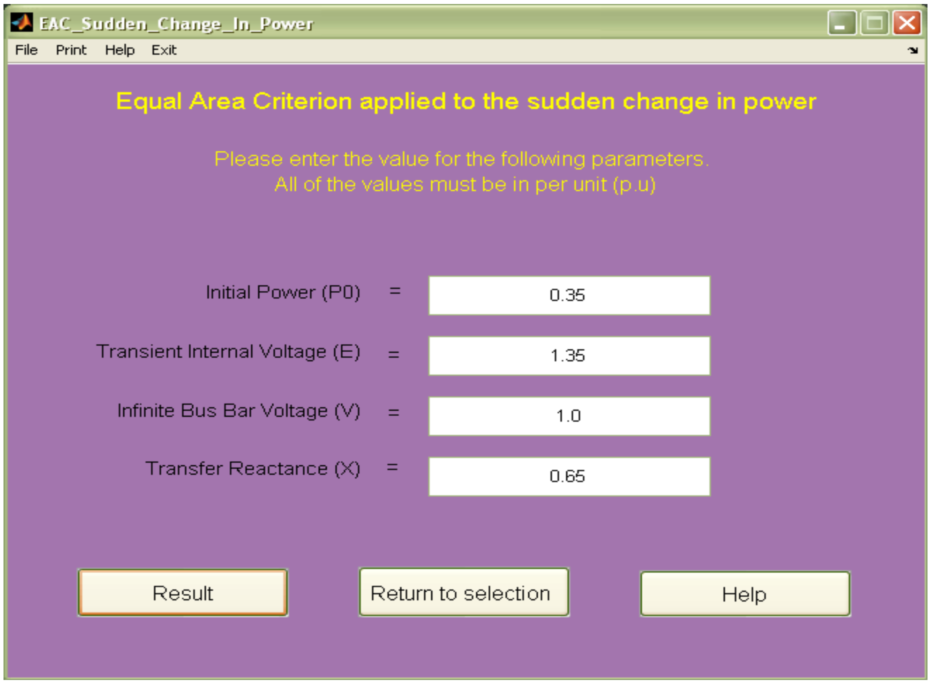


Figure 4.30 Application of Equal Area Criterion due to sudden change in Power

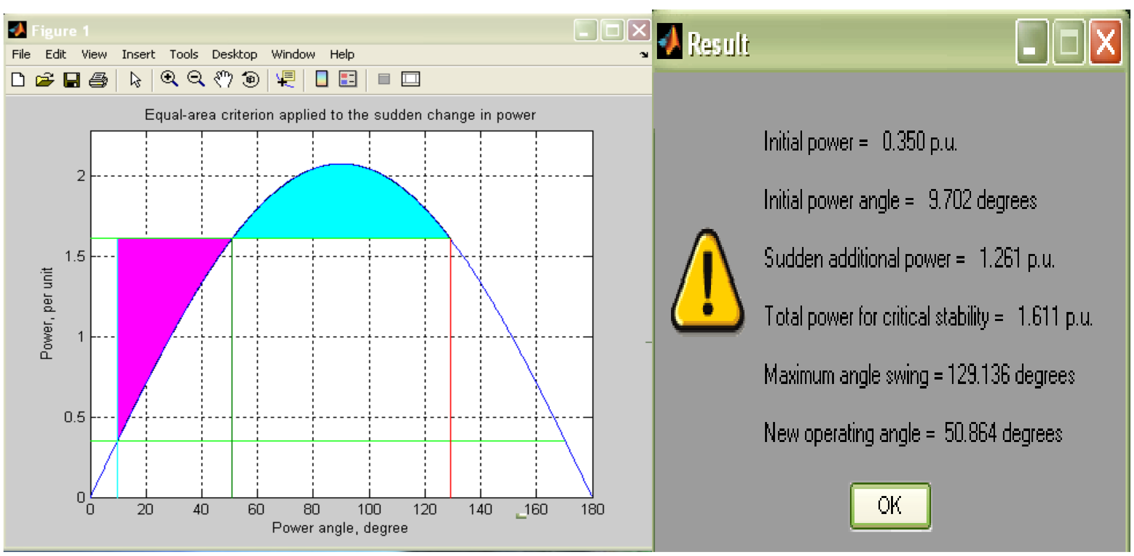


Figure 4.31 Result on Application of Equal Area Criterion due to sudden change in Power

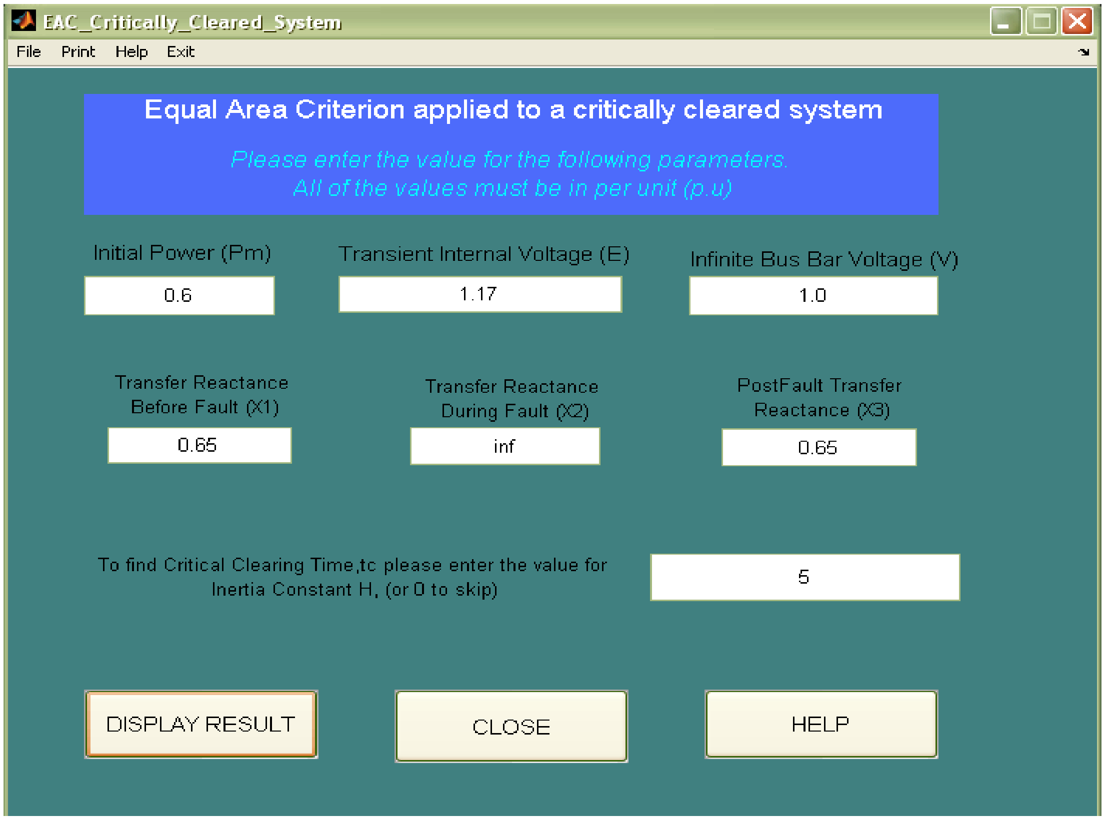


Figure 4.32 Application of Equal Area Criterion due to Critically Cleared System

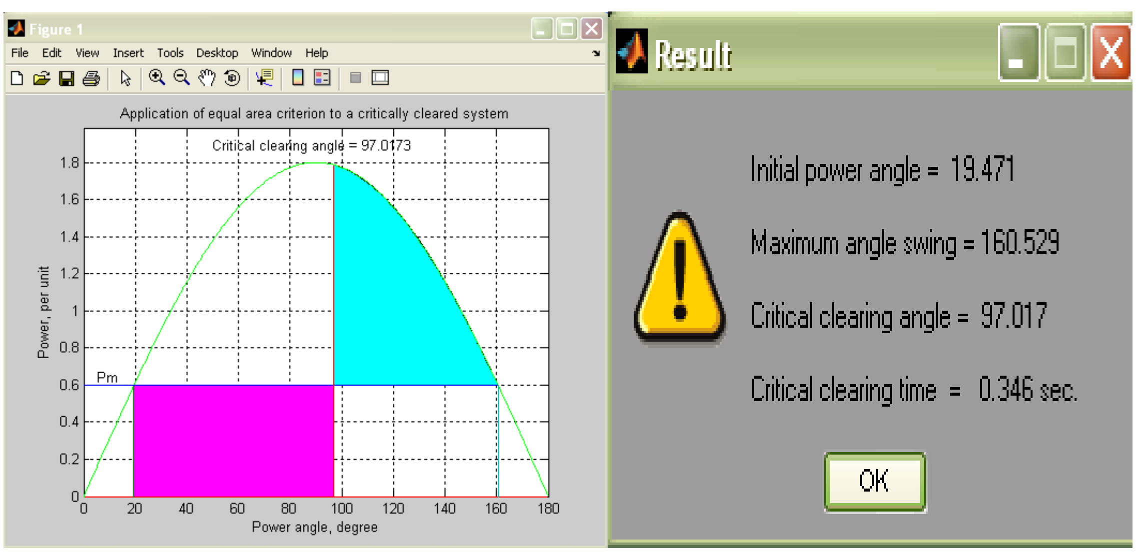


Figure 4.33 Result on Application of Equal Area Criterion due to Critically Cleared System

4.6 User Guide On How To Use Power System Analysis Educational Toolbox (Optimal Dispatch of Power Generation)

The main layout of optimal dispatch is in Figure 4.10. Four pushbuttons have been built in this analysis. There are *Definition of Optimal Dispatch*, *Example and Tutorial*, *Begin the Analysis* and *Exit* pushbutton. Definition pushbutton of Optimal Dispatch is shown in Figure 4.34.

The Example and Tutorial pushbutton is chosen to view the tutorial that provided in the toolbox. The layout can be seen in Figure 4.35. On this window, it will discuss about Data Preparation, simulation on Neglecting Losses and No Generator Limit, simulation on Neglecting Losses and Including Generator Limit, simulation on Optimal Dispatch Including losses, and lastly is Optimal Dispatch with Power Flow solution. The layout can be seen in Figure 4.36 to Figure 4.2 repectively.

Figure 4.38 is show on how user must fill in the input data for problem in Figure 4.37. For this input data, user must fill in fuel cost and total load of power system. To view back the question for this problem, user needs to click the *Show* button at the top of this window meanwhile *Solve* button will simulate the result of this analysis.

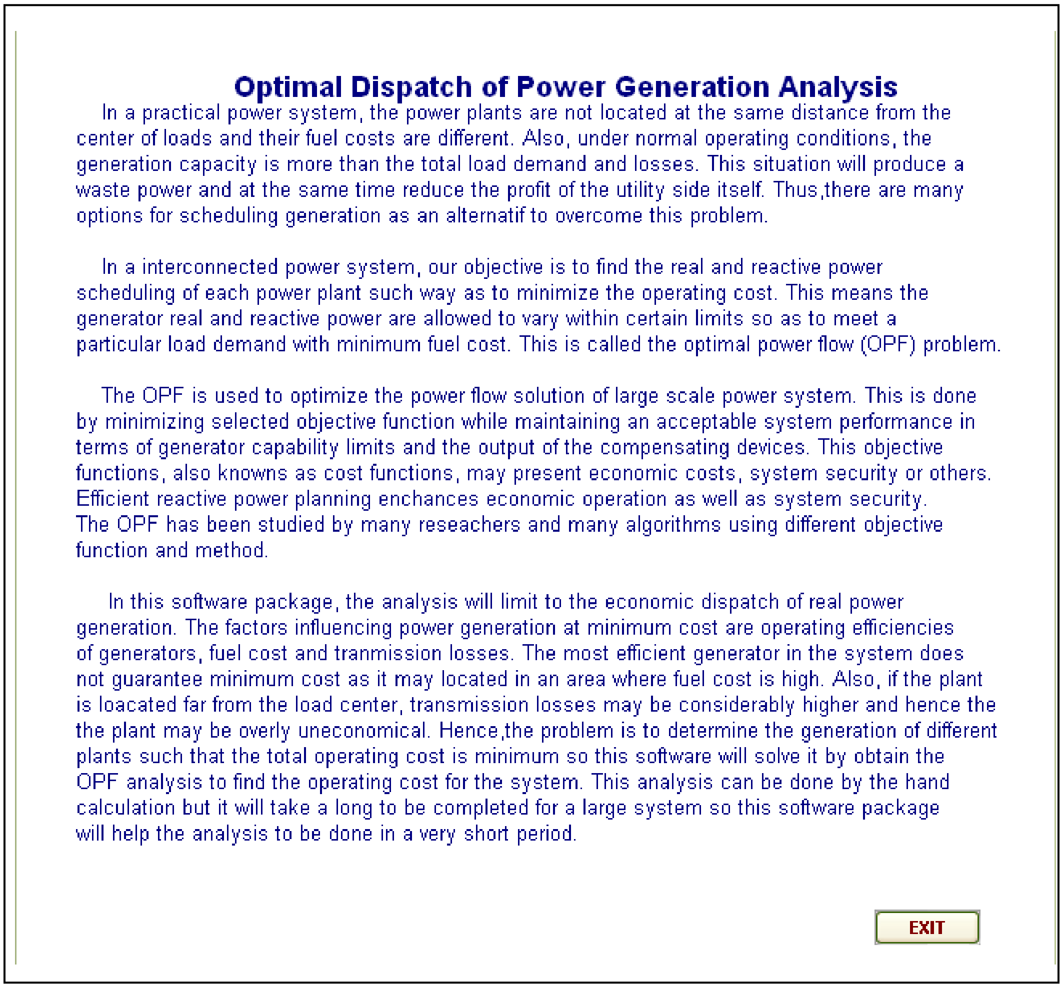


Figure 4.34 Definition of Optimal Dispatch

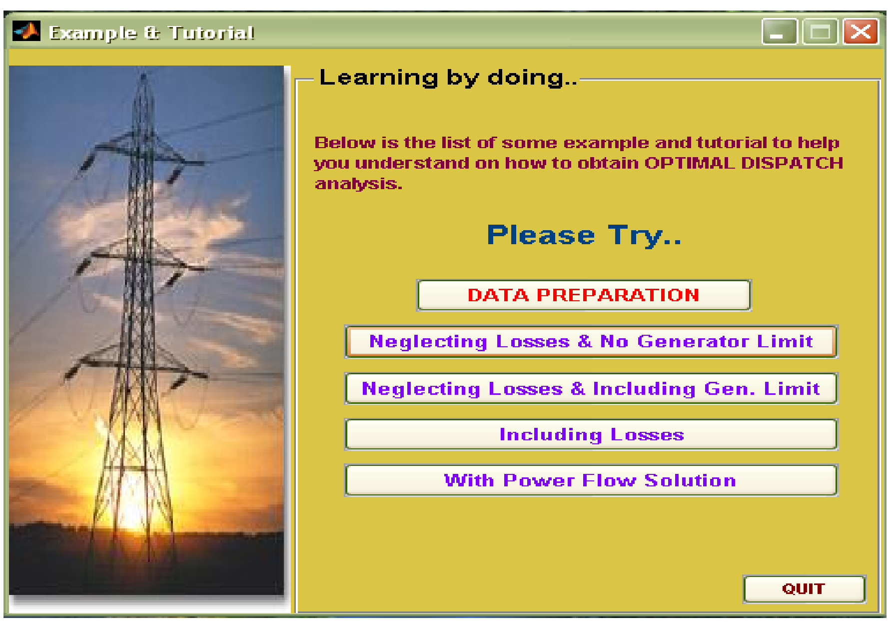


Figure 4.35 Example and Tutorial Layout

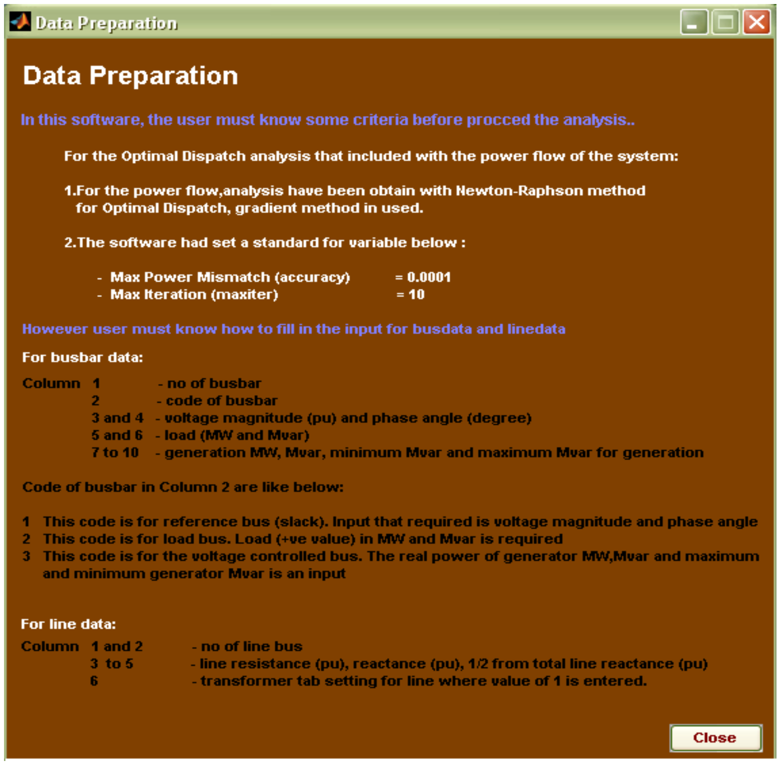


Figure 4.36 Data preparation

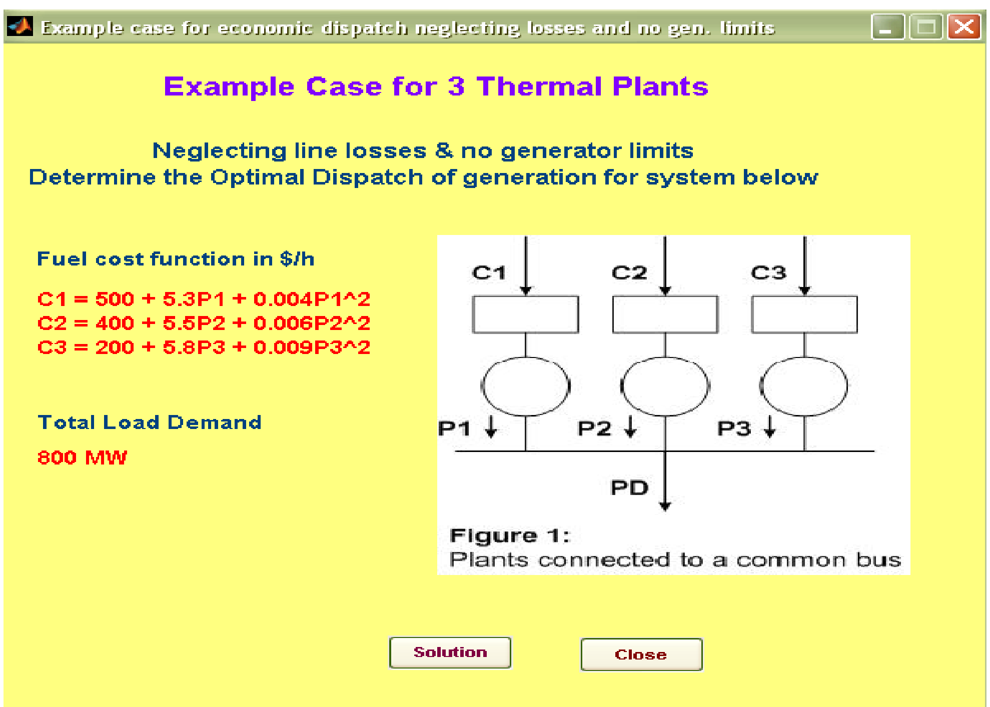


Figure 4.37 Optimal Dispatch on Neglecting Losses and No Generator Limit

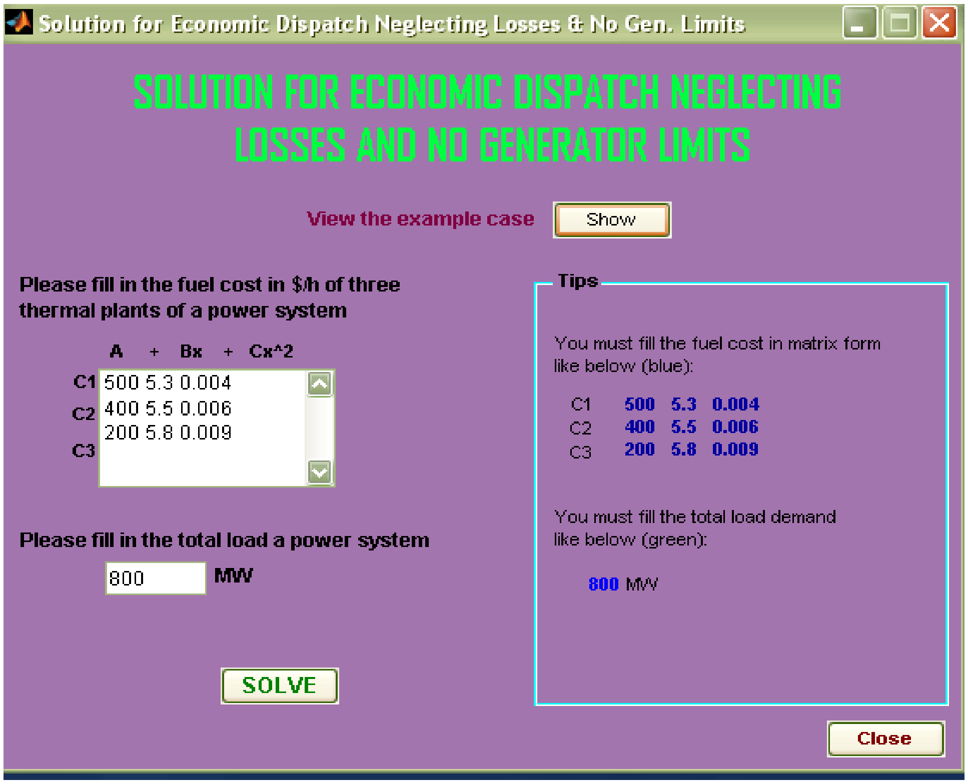


Figure 4.38 Input Data for Optimal Dispatch Neglecting Losses and No Generator Limit

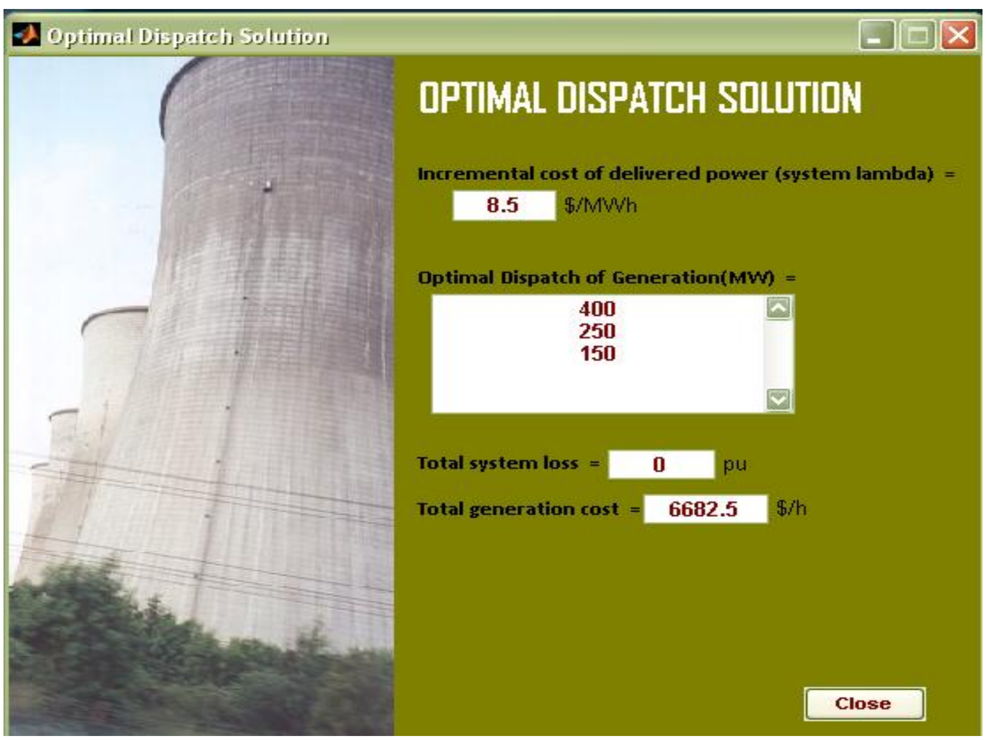


Figure 4.39 Result for Optimal Dispatch Neglecting Losses and No Generator Limit

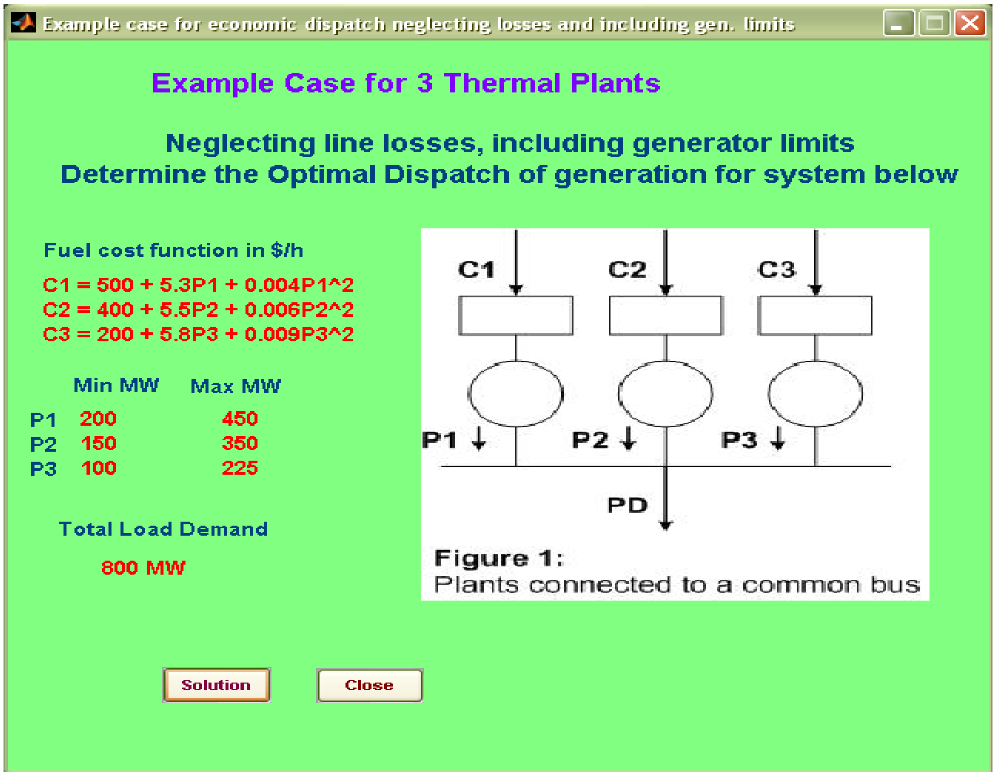


Figure 4.40 Optimal Dispatch on Neglecting Losses and Include Generator Limit

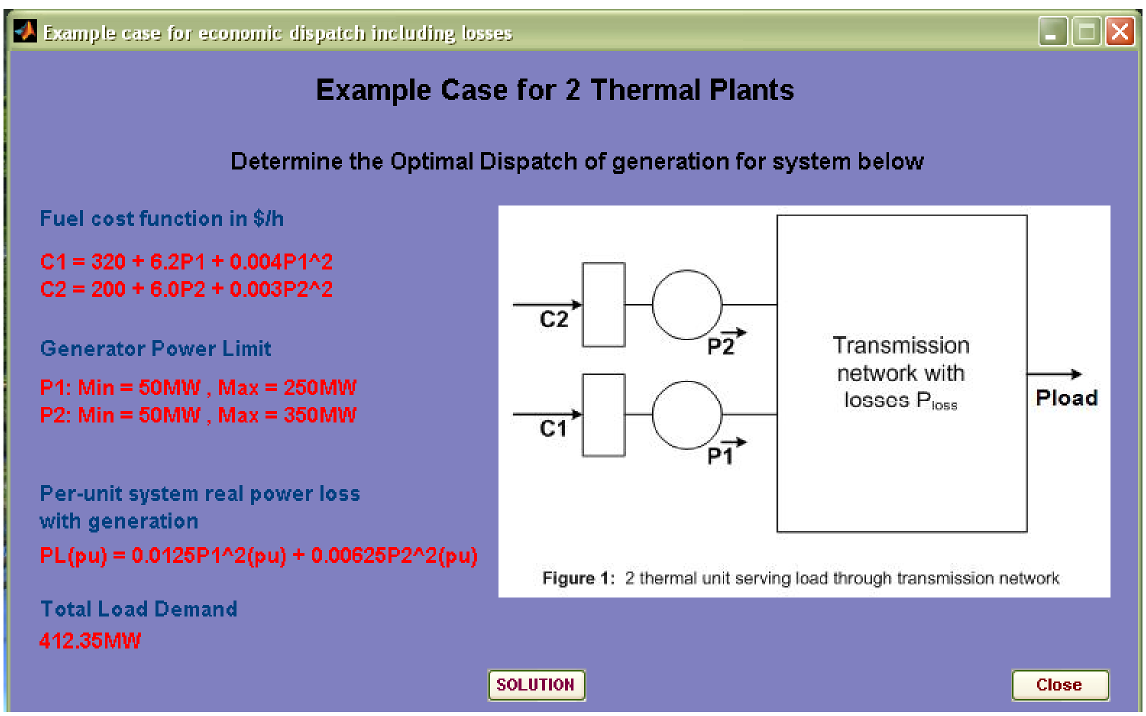


Figure 4.41 Optimal Dispatch Include Losses

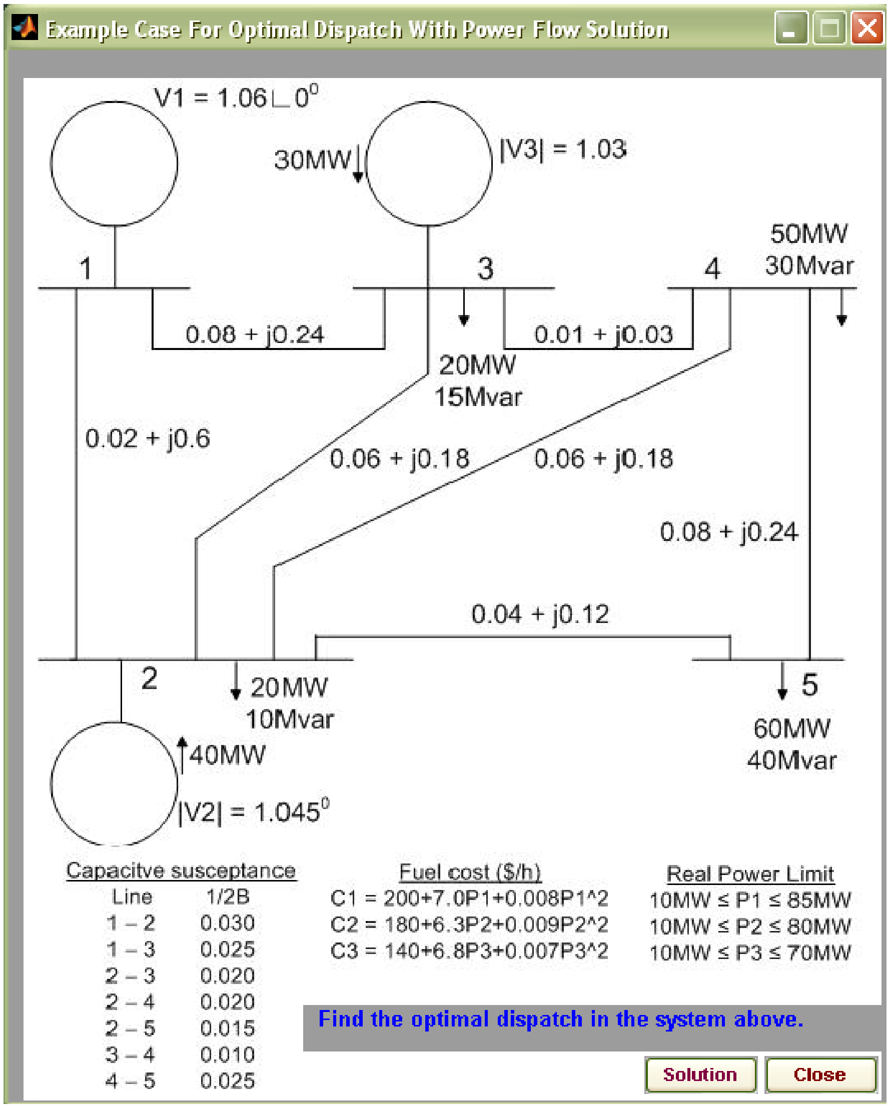


Figure 4.42 Optimal Dispatch with Power Flow solution

4.7 Discussion on Power System Analysis Educational Toolbox

In this part, the author will discuss on the simulation that had be done using this educational toolbox.

4.7.1 Discussion on Power Flow Analysis Simulation

This sub-chapter will focus on the result of simulation 3-bus system by using Newton-Raphson method, Gauss-Seidel method and Fast Decoupled method. The diagram of this system has been shown in Figure 4.14.

4.7.1.1 3-bus System Analysis

In order to make this discussion more easier to see, the author will make all the simulation in table form.

Table 4.1 : Newton-Raphson Method

Bus No	Voltage Magnitude	Angle Degree	Load		Generation		Injected
			MW	MVar	MW	MVar	MVar
1	1.050	0.000	0.000	0.000	221.270	109.269	0.000
2	0.972	-3.519	400.000	250.000	0.000	0.000	0.000
3	1.040	-1.839	2.400	1.200	200.000	180.083	0.000

Table 4.2 : Gauss-Seidel Method

Bus No	Voltage Magnitude	Angle Degree	Load		Generation		Injected
			MW	MVar	MW	MVar	MVar
1	1.050	0.000	0.000	0.000	221.235	109.135	0.000
2	0.972	-3.517	400.000	250.000	0.000	0.000	0.000
3	1.040	-1.838	2.400	1.200	200.000	180.151	0.000

Table 4.3 : Fast- Decoupled Method

Bus No	Voltage Magnitude	Angle Degree	Load		Generation		Injected
			MW	MVar	MW	MVar	MVar
1	1.050	0.000	0.000	0.000	221.235	109.135	0.000
2	0.972	-3.517	400.000	250.000	0.000	0.000	0.000
3	1.040	-1.838	2.400	1.200	200.000	180.151	0.000

From Table 4.1, Table 4.2 and Table 4.3, the result on voltage magnitude, angle degree, load and generation are quiet same for all analysis method.

The line flow and line losses for 3 bus power system network is tabulated in Table 4.4.

Table 4.4 : Newton-Raphson Method

Line		Power at bus and Line Flow			Line Losses		Transformer
From	To	MW	MVar	MVA	MW	MVar	Tap
1		221.270	109.269	246.779	-	-	-
	2	209.510	105.979	234.789	10.000	20.000	-
	3	11.790	3.294	12.241	0.014	0.408	-
2		-400.000	-250.000	471.699	-	-	-
	1	-199.510	-85.978	217.248	10.000	20.000	-
	3	-200.490	-164.022	259.035	8.886	17.772	-
3		197.600	178.883	266.542	-	-	-
	1	-11.776	-2.887	12.125	0.014	0.408	-
	2	209.376	181.793	277.285	8.886	17.772	-

Table 4.5 : Gauss-Seidel Method

Line		Power at bus and Line Flow			Line Losses		Transformer
From	To	MW	MVar	MVA	MW	MVar	Tap
1		221.235	109.135	246.689	-	-	-
	2	209.479	106.033	234.786	10.000	20.000	-
	3	11.784	3.294	12.236	0.014	0.407	-
2		-400.000	-250.000	471.699	-	-	-
	1	-199.480	-86.033	217.241	10.000	20.000	-
	3	-200.489	-164.084	259.079	8.889	17.778	-
3		197.600	178.951	266.588	-	-	-
	1	-11.771	-2.887	12.120	0.014	0.407	-
	2	209.378	181.861	277.331	8.889	17.778	-

Table 4.6 : Fast- Decoupled Method

Line		Power at bus and Line Flow			Line Losses		Transformer
From	To	MW	MVar	MVA	MW	MVar	Tap
1		221.285	109.275	246.796	-	-	-
	2	209.508	105.968	234.783	10.000	19.999	-
	3	11.790	3.294	12.241	0.014	0.408	-
2		-400.000	-250.000	471.699	-	-	-
	1	-199.509	-85.969	217.242	10.000	19.999	-
	3	-200.486	-164.007	259.023	8.885	17.770	-
3		197.600	178.914	266.563	-	-	-
	1	-11.776	-2.887	12.125	0.014	0.408	-
	2	209.371	181.776	277.271	8.885	17.770	-

The summary of this line flow and losses can be seen in Table 4.7 below.

Table 4.7 : Summary of Line Flow and Losses

Method	Total Generation (MVA)	Total Load (MVA)	Total Loss (MVA)
Newton-Raphson	421.270 + j289.352	402.4 + j251.2	18.900 + j38.180
Gauss-Seidel	421.235 + j289.286	402.4 + j251.2	18.902 + j38.185
Fast Decouple	421.285 + j289.389	402.4 + j251.2	18.898 + j38.177

From the Power System Analysis Pro Evolution Toolox, when Newton Raphson method is applied to find maximum power mismatch, its shows the maximum power mismatch is equal to 0.000266997 and the number of iteration is equal to 3. Meanwhile, if Gauss-Seidel is used, the maximum power mismatch is become larger that are, 0.000704109 and the number of iteration is 5. If the Fast decoupled method is applied, the result will display maximum power mismatch of 0.000455801 and number of iteration is equal to 13. This be simplified in Table 4.8 below.

Table 4.8 : Maximum Power mismatch

Method	No of Iteration	Maximum Power Mismatch	Total Loss (MVA)
Newton-Raphson	3	0.000266997	18.900 + j38.180
Gauss-Seidel	5	0.000704109	18.902 + j38.185
Fast Decouple	13	0.000455801	18.898 + j38.177

4.7.2 Discussion on Fault Analysis Simulation

This sub-chapter will discuss on Fault Analysis. There are four types fault analysis that will be discuss that are, *Three Phase Balanced Fault Analysis, Single Line to Ground Fault Analysis, Line to Line Fault Analysis,* and *Double Line to Ground Fault Analysis.* This analysis will cover on 3-bus network.

4.7.2.1 Result of Simulation (3-bus network)

The diagram of 3-bus network is be shown in Figure 4.24. To solve this problem, user need to find bus impedance for positive, negative and zero sequence impedance (Zbus0). Positive and negative bus impedance (Zbus1/Zbus2) are same. The bus impedance data must be entered correctly as in the Figure 4.25. Then the fault location and fault impedance is entered to get the result. The bus impedance for 3 bus network is shown in Table 4.9, Table 4.10, and Table 4.11.

Table 4.9 : Positive/Negative Bus Impedance (Zbus1/Zbus2)

0 + 0.145i	0 + 0.105i	0 + 0.13i
0 + 0.105i	0 + 0.145i	0 + 0.12i
0 + 0.130i	0 + 0.120i	0 + 0.22i

Table 4.10 : Zero Bus Impedance (Zbus0)

0 + 0.1820i	0 + 0.054500i	0 + 0.140i
0 + 0.0545i	0 + 0.086375i	0 + 0.065i
0 + 0.1400i	0 + 0.065000i	0 + 0.350i

Table 4.11 : Fault Location and Impedance

Fault Location (Bus No)	3
Fault Impedance (p.u)	0.1i

The bus voltage and line current during fault at bus 3 is tabulated in Table 4.12 until Table 4.15.

Table 4.12 : Three Phase Balanced Fault Analysis

Fault Impedance (Real)			0
Fault Impedance (Imaginary)*j			1.000000e-001
Total fault current			3.1250 per unit
Bus Voltages during fault in per unit			
Bus No	Voltage Magnitude (p.u)		Angle (degrees)
1	0.5938		0
2	0.6250		0
3	0.3125		0
Line currents for fault at bus No. 3			
From Bus	To Bus	Current magnitude (p.u)	Angle (degrees)
G	1	1.6250	-90
1	3	1.8750	-90
G	2	1.5000	-90
2	1	0.2500	-90
2	3	1.2500	-90
3	F	3.1250	-90

Table 4.13 : Single Line to Ground Fault Analysis

Fault Impedance (Real)				0
Fault Impedance (Imaginary)*j				1.000000e-001
Total fault current				3.1250 per unit
Bus Voltages during fault in per unit				
Bus No		Voltage Magnitude (p.u)		
		Phase a	Phase b	Phase c
1		0.6330	1.0046	1.0046
2		0.7202	0.9757	0.9757
3		0.2752	1.0647	1.0647
Line currents for fault at bus No. 3				
From Bus	To Bus	Line Current magnitude (p.u)		
		Phase a	Phase b	Phase c
1	3	1.6514	0.0000	0.0000
2	1	0.3761	0.1560	0.1560
2	3	1.1009	0.0000	0.0000
3	F	2.7523	0.0000	0.0000

Table 4.14 : Line to Line Fault Analysis

Fault Impedance (Real)			0	
Fault Impedance (Imaginary)*j			1.000000e-001	
Total fault current			3.2075 per unit	

Bus Voltages during fault in per unit

Bus No	Voltage Magnitude (p.u)		
	Phase a	Phase b	Phase c
1	1.0000	0.6720	0.6720
2	1.0000	0.6939	0.6939
3	1.0000	0.5251	0.5251

Line currents for fault at bus No. 3

From Bus	To Bus	Line Current magnitude (p.u)		
		Phase a	Phase b	Phase c
1	3	0.0000	1.9245	1.9245
2	1	0.0000	0.2566	0.2566
2	3	0.0000	1.2830	1.2830
3	F	0.0000	3.2075	3.2075

Table 4.15 : Line to Line Fault Analysis

Fault Impedance (Real)	0
Fault Impedance (Imaginary)*j	1.000000e-001
Total fault current	1.9737 per unit

Bus Voltages during fault in per unit

Bus No	Voltage Magnitude (p.u)		
	Phase a	Phase b	Phase c
1	1.0066	0.5088	0.5088
2	0.9638	0.5740	0.5740
3	1.0855	0.1974	0.1974

Line currents for fault at bus No. 3

From Bus	To Bus	Line Current magnitude (p.u)		
		Phase a	Phase b	Phase c
1	3	0.0000	2.4350	2.4350
2	1	0.1118	0.3682	0.3682
2	3	0.0000	1.6233	1.6233
3	F	0.0000	4.0583	4.0583

4.7.2.2 Analysis of Fault Analysis

The simulation on 11-bus network and 32-bus network is done in order to make comparing to find the most severe damage that could effect the system. The result is tabulated in table 4.16 below.

Table 4.16 : Result of 3 bus network, 11 bus network and 32 bus network

Bus Network	Balanced 3 Phase Fault (p.u)	Single line to Ground Fault (p.u)	Double Line to Ground fault (p.u)	Line to Line Fault (p.u)
3	3.125	2.7523	1.9737	3.2075
11	3.1392	2.7230	2.4043	2.7186
32	5.4719	2.9128	1.9806	4.7388

From the table, author can conclude that, the most severe damage to transmission line is balanced three phase fault. It is because, balanced three phase fault produces highest rated current other than single line to ground fault, double line to ground fault and line to line fault.

4.7.3 Discussion on Stability Analysis

The Stability Analysis module will cover on *Steady-state, Equal Area Criterion* and *Multimachine Transient Stability*.

4.7.3.1 Steady-State stability

Steady state stability is the ability of the system to remain synchronism after having small and slow disturbances. Lets say we have a one line diagram as Figure 4.43 as below:

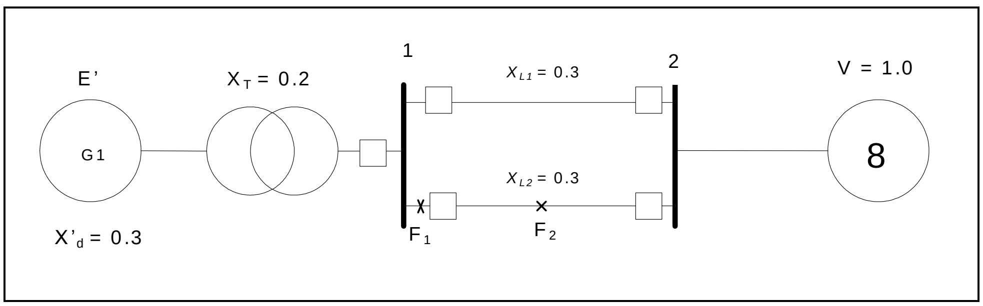


Figure 4.43 One line diagram for steady state stability

The excitation voltage, E and transfer reactance, X must be obtained from formula calculation and circuit analysis before begin the simulation. All the input data is given in Table 4.17 below.

Table 4.17 : Input Data for Steady-State Analysis

Inertia Constant, H	9.94 MJ/MVA
Transient Reactance, X_d	0.3 p.u
Real power, P_m	0.6 p.u
Infinite bus voltage, V	1.0 p.u
Damping Power Coefficient, D	0.138
Frequency, f_o	50 Hz
Excitation Voltage, E	1.35
Transfer reactance, X	0.65

We get the result as Figure 4.44 as below

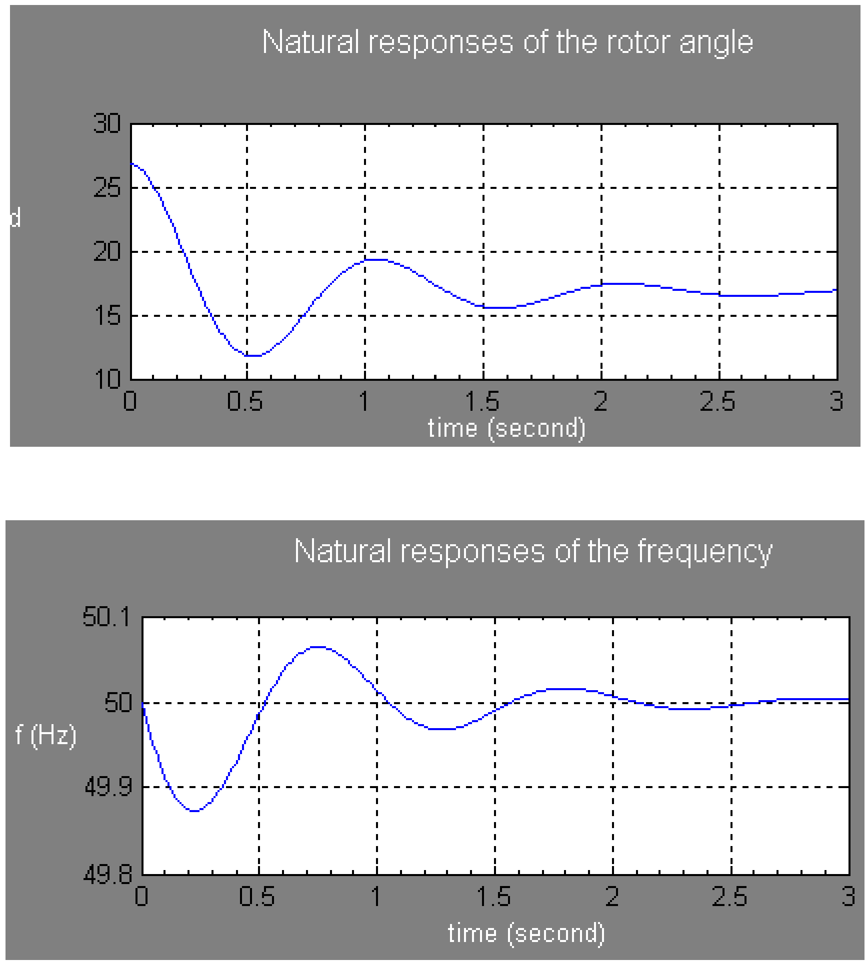


Figure 4.44 Natural response of the rotor angle and frequency for Steady State Stability

The response shows a small disturbance will be followed by a relatively slowly damped oscillation or swing of the rotor, before steady state operation at synchronous speed is resumed. In the result above, the response is about $t_s \approx 4r = 4\ (1/1.3) \approx 3.1$ s. We also observe that the oscillation is fairly low in frequency, in the order of 0.955Hz.

4.7.3.2 Equal Area Criterion

Equal Area criterion can be used for quick prediction of stability.

Application of Equal Area Criterion due to sudden change in Power

In this simulation, previous example has been chosen to discuss the Steady state stability analysis. All the input data is given in Table 4.18.

Table 4.18 : Input Data of Equal Area Criterion due to sudden change in Power

Real power, P_m	0.6 p.u
Infinite bus voltage, V	1.0 p.u
Excitation Voltage, E	1.35
Transfer reactance, X	0.65

The result is shown in Table 4.19 and Figure 4.45

Table 4.19 : Result of Equal Area Criterion due to sudden change in Power

Parameter	Value (p.u)
Initial Power	0.6
Initial Power Angle	16.791 degree
Sudden additional Power	1.084
Total Power for Critical stability	1.684
Maximum angle swing	125.840 degree
New Operating angle	54.160 degree

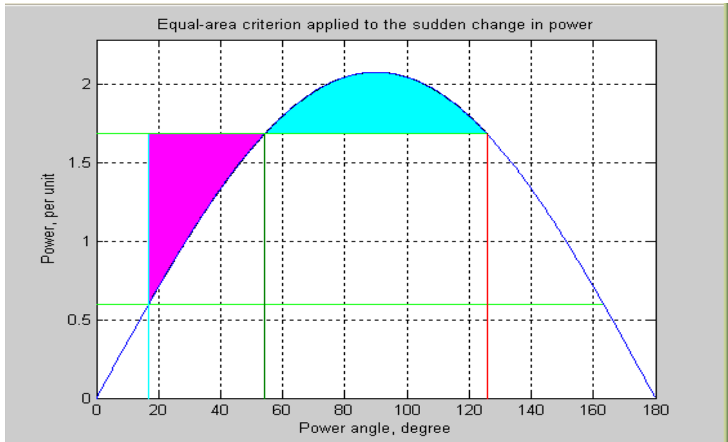


Figure 4.45 Result of Equal Area Criterion due to sudden change in Power

The Equal Area Criterion is used to determine the maximum additional power Pm which can be applied for stability to be maintained. With a sudden change in the power input, the stability is maintain only if area A2 (green) at least equal to A1(pink) can be located above Pm. If the area A2 < A1, the accelerating momentum can never be overcome. The limit of the stability occurs when δmax is at the intersection line Pm and the power angle curve for 90° < δ < 180°.

At the result above, before disturbance, the machine operating at the equilibrium point, δ_0 (16.791°) with mechanical power input, is equal to electrical power output ($P_{m0} = P_{e0}$) that is 0.6 p.u. When a sudden increase in input power, mechanical power input become increase to 1.684 p.u. Since mechanical power input increase, it has a large power than electrical power output ($P_{m1} > P_{e0}$), the rotor will accelerate and power angle increase to 54.160°. This is covered in Area 1 (pink color on graph).

When power angle become 54.160°, at the same time electrical power will equal to P_{m1} and make the rotor once again running above synchronous speed and thus δ become increase to 125.840°. When its happen, mechanical power will less than electrical power and make the rotor decelerate toward synchronous speed. This will produce Area 2 (green color on graph).

The power system will maintain its stability if Area 1 is equal to area 2. But if the Area 1 is larger than Area 2, the system will loss stability.

4.8 Conclusion

In Power Flow Analysis, there are 3 method of solution that have been discussed in this toolbox that are, Newton Raphson method, Gauss-Seidel Method and Fast Decoupled method. From all this methods, the best method to solve problem in Power Flow Analysis is Newton Raphson method because, it has low power mismatch and it was close to zero.

In Fault Analysis, 4 types of fault been discussed that are, Balanced Three Phase Fault, Single Line to Ground Fault Analysis, Line to Line Fault Analysis and Double Line to Ground Fault Analysis. From the result that shown before, the most severe damage to transmission line is Balanced Three Phase Fault. It is because, Balanced Three Phase Fault will produces highest rated current other than Single Line to Ground Fault Analysis, Line to Line Fault Analysis and Double Line to Ground Fault Analysis.

In Stability Analysis, two important things that must be considered are critical clearing time, t_c and critical clearing angle, δ_c. Critical clearing time is the time for power angle reach the critical clearing angle, and critical clearing angle is reached when δ_{max} is at intersection of P_m and P_e. The circuit breaker must be open the faulted line before δ reach the critical clearing angle.

Optimal Dispatch is used to optimize the power flow solution large scale power system. This is done by minimizing selected objective functions while maintaining an acceptable system performance in terms of generator capability limits and the output of the compensating devices.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

At the end of this project, a prototype of Power System Analysis Educational toolbox has been developed. The development of this toolbox is presented in detail using Matlab 7.1. The development of GUI was quite easy to design, but the difficult part is, to program the source code and command which needs more practices, reading and effort.

The objectives of this project which are to develop educational toolbox of Power System Analysis and to obtain simulation and analysis of Power System Analysis was successfully achieved.

5.2 Future Recommendation

For the improvement of this toolbox, the author would suggest to upgrade this project in order to makes it becomes more interesting and user-friendly.

Firstly the author would like to suggest the new developer to make this software to run in one GUI window only. This will avoid many GUI window appear at same time. By doing so, this program can be run in one window only which helps the user to use this program easier.

Secondly, graphic pictures or 3-dimensional images can be added to this program to make it more matured and interesting. While the author develops this project, the author only uses the image at the Property Inspector in Matlab 7.1 GUI.

Thirdly, the author likes to recommend to modify this educational toolbox into other language such as Bahasa Malaysia. By doing so, who could not understand English can also use this program. Currently, this toolbox was developed totally in English.

Besides that, the combination between Matlab Simulink and GUI can be created in order to built more interactive and useful toolbox for the user.

Lastly, the author also wants to recommend to add more Power System Analysis problems such as Transmission Line Parameter and Line Model and performance to this toolbox to make it more useful and functional to the lecturers and students.

REFERENCES

[1] John J. Grainger, William D. Stevenson, *Power System Analysis*, International Edition, McGraw-Hill, 1994

[2] 10th January, 2008, Citing Internet Sources, URL http://en.wikipedia.org/wiki/Gauss-Seidel_Method

[3] 10th January, 2008, Citing Internet Sources, URL http://en.wikipedia.org/wiki/Newton-Raphson_Method

[4] 10th January, 2008, Citing Internet Sources, URL http://en.wikipedia.org/wiki/Jacobian_matrix

[5] Hadi Saadat, *Power System Analysis*, McGraw-Hill, International Edition, 1999.

[6] Allen J. Wood, Power Generation, Operation and Control, John Wiley & Sons Inc, Second Edition, 1996

[7] 10th January, 2008, Citing Internet Sources, URL http://en.wikipedia.org/wiki/GUI

[8] 10th January, 2008, Citing Internet Sources, URL www.ewh.ieee.org/r8/uae/GUI.pdf

APPENDIX A: RESULT OF IEEE 30-BUS POWER SYSTEM

Line Flow and Losses

--Line-- from	to	Power at bus & line flow MW	Mvar	MVA	--Line loss-- MW	Mvar	Transformer tap
1		260.998	-17.021	261.553			
	2	177.778	-22.148	179.152	5.464	10.524	
	3	83.221	5.127	83.378	2.808	7.085	
2		18.300	36.122	40.493			
	1	-172.314	32.671	175.384	5.464	10.524	
	4	45.712	2.705	45.792	1.106	-0.517	
	5	82.990	1.703	83.008	2.995	8.178	
	6	61.912	-0.958	61.920	2.048	2.264	
3		-2.400	-1.200	2.683			
	1	-80.412	1.958	80.436	2.808	7.085	
	4	78.012	-3.158	78.076	0.771	1.344	
4		-7.600	-1.600	7.767			
	2	-44.605	-3.222	44.722	1.106	-0.517	
	3	-77.242	4.503	77.373	0.771	1.344	
	6	70.126	-17.526	72.282	0.604	1.179	
	12	44.121	14.646	46.489	0.000	4.685	0.932
5		-94.200	16.975	95.717			
	2	-79.995	6.475	80.257	2.995	8.178	
	7	-14.205	10.500	17.664	0.151	-1.687	
6		0.000	0.000	0.000			
	2	-59.864	3.222	59.951	2.048	2.264	
	4	-69.521	18.705	71.994	0.604	1.179	
	7	37.523	-1.885	37.570	0.367	-0.598	
	8	29.528	-3.754	29.766	0.103	-0.558	
	9	27.693	-7.322	28.644	-0.000	1.594	0.978
	10	15.823	0.653	15.836	0.000	1.278	0.969
	28	18.819	-9.618	21.134	0.060	-13.086	

7		-22.800	-10.900	25.272		
	5	14.356	-12.187	18.831	0.151	-1.687
	6	-37.156	1.287	37.178	0.367	-0.598
8		-30.000	0.826	30.011		
	6	-29.425	3.196	29.598	0.103	-0.558
	28	-0.575	-2.370	2.438	0.000	-4.368
9		0.000	0.000	0.000		
	6	-27.693	8.916	29.093	-0.000	1.594
	11	0.000	-15.657	15.657	0.000	0.462
	10	27.693	6.741	28.501	0.000	0.809
10		-5.800	17.000	17.962		
	6	-15.823	0.626	15.835	0.000	1.278
	9	-27.693	-5.932	28.321	0.000	0.809
	20	9.027	3.560	9.704	0.081	0.180
	17	5.372	4.414	6.953	0.014	0.037
	21	15.733	9.842	18.558	0.110	0.237
	22	7.583	4.490	8.813	0.052	0.107
11		0.000	16.119	16.119		
	9	-0.000	16.119	16.119	0.000	0.462
12		-11.200	-7.500	13.479		
	4	-44.121	-9.961	45.232	0.000	4.685
	13	0.000	-10.291	10.291	0.000	0.133
	14	7.856	2.442	8.227	0.075	0.155
	15	17.857	6.947	19.161	0.217	0.428
	16	7.208	3.363	7.954	0.053	0.112
13		0.000	10.423	10.423		
	12	-0.000	10.424	10.424	0.000	0.133
14		-6.200	-1.600	6.403		
	12	-7.782	-2.287	8.111	0.075	0.155
	15	1.582	0.687	1.724	0.006	0.005
15		-8.200	-2.500	8.573		
	12	-17.640	-6.519	18.806	0.217	0.428
	14	-1.576	-0.681	1.717	0.006	0.005
	18	6.014	1.744	6.262	0.039	0.080
	23	5.001	2.956	5.810	0.031	0.063

16		-3.500	-1.800	3.936		
	12	-7.154	-3.251	7.858	0.053	0.112
	17	3.654	1.451	3.932	0.012	0.027
17		-9.000	-5.800	10.707		
	16	-3.643	-1.424	3.911	0.012	0.027
	10	-5.357	-4.376	6.918	0.014	0.037
18		-3.200	-0.900	3.324		
	15	-5.975	-1.665	6.203	0.039	0.080
	19	2.775	0.765	2.879	0.005	0.010
19		-9.500	-3.400	10.090		
	18	-2.770	-0.755	2.871	0.005	0.010
	20	-6.730	-2.645	7.231	0.017	0.034
20		-2.200	-0.700	2.309		
	19	6.747	2.679	7.259	0.017	0.034
	10	-8.947	-3.379	9.564	0.081	0.180
21		-17.500	-11.200	20.777		
	10	-15.623	-9.606	18.340	0.110	0.237
	22	-1.877	-1.594	2.462	0.001	0.001
22		0.000	0.000	0.000		
	10	-7.531	-4.384	8.714	0.052	0.107
	21	1.877	1.596	2.464	0.001	0.001
	24	5.654	2.788	6.304	0.043	0.067
23		-3.200	-1.600	3.578		
	15	-4.970	-2.893	5.751	0.031	0.063
	24	1.770	1.293	2.192	0.006	0.012
24		-8.700	-2.400	9.025		
	22	-5.611	-2.721	6.236	0.043	0.067
	23	-1.764	-1.280	2.180	0.006	0.012
	25	-1.325	1.602	2.079	0.008	0.014
25		0.000	0.000	0.000		
	24	1.333	-1.588	2.073	0.008	0.014
	26	3.545	2.366	4.262	0.045	0.066
	27	-4.877	-0.778	4.939	0.026	0.049

25		0.000	0.000	0.000			
	24	1.333	-1.588	2.073	0.008	0.014	
	26	3.545	2.366	4.262	0.045	0.066	
	27	-4.877	-0.778	4.939	0.026	0.049	
26		-3.500	-2.300	4.188			
	25	-3.500	-2.300	4.188	0.045	0.066	
27		0.000	0.000	0.000			
	25	4.903	0.827	4.972	0.026	0.049	
	28	-18.184	-4.157	18.653	0.000	1.309	
	29	6.189	1.668	6.410	0.086	0.162	
	30	7.091	1.661	7.283	0.161	0.304	
28		0.000	0.000	0.000			
	27	18.184	5.466	18.987	0.000	1.309	0.968
	8	0.575	-1.999	2.080	0.000	-4.368	
	6	-18.759	-3.467	19.077	0.060	-13.086	
29		-2.400	-0.900	2.563			
	27	-6.104	-1.506	6.286	0.086	0.162	
	30	3.704	0.606	3.753	0.033	0.063	
30		-10.600	-1.900	10.769			
	27	-6.930	-1.358	7.062	0.161	0.304	
	29	-3.670	-0.542	3.710	0.033	0.063	
Total loss					17.599	22.244	

APPENDIX B: POWER FLOW SOLUTION OF IEEE 30-BUS

Maximum Power Mismatch = 7.54898e-007

No of Iteration = 4

Bus No	Voltage Magnitude	Angle Degree	Load		Generation		Injected
			MW	MVar	MW	MVar	MVar
1	1.060	0.0000	0.0	0.0	260.998	-17.0209	0.000
2	1.043	-5.4968	21.7	12.7	40	48.8221	0.000
3	1.0215	-8.0040	2.400	1.200	0.000	0.0000	0.000
4	1.01285	-9.6614	7.6	1.6	0	0	0
5	1.01	-14.381	94.2	19	0	35.9746	0
6	1.01206	-11.398	0	0	0	0	0
7	1.00345	-13.149	22.8	10.9	0	0	0
8	1.01	-12.115	30	30	0	30.8264	0
9	1.05101	-14.433	0	0	0	0	0
10	1.04436	-16.024	5.8	2	0	0	19
11	1.082	-14.433	0	0	0	16.1185	0
12	1.05737	-15.302	11.2	7.5	0	0	0
13	1.071	-15.302	0	0	0	10.4235	0
14	1.04244	-16.191	6.2	1.6	0	0	0
15	1.03778	-16.278	8.2	2.5	0	0	0
16	1.04467	-15.880	3.5	1.8	0	0	0
17	1.03913	-16.188	9	5.8	0	0	0
18	1.02794	-16.883	3.2	0.9	0	0	0
19	1.02526	-17.051	9.5	3.4	0	0	0
20	1.02925	-16.852	2.2	0.7	0	0	0
21	1.03209	-16.468	17.5	11.2	0	0	0
22	1.03267	-16.454	0	0	0	0	0
23	1.02723	-16.662	3.2	1.6	0	0	0
24	1.02156	-16.830	8.7	6.7	0	0	4.3
25	1.01887	-16.423	0	0	0	0	0
26	1.00122	-16.842	3.5	2.3	0	0	0
27	1.02574	-15.912	0	0	0	0	0
28	1.01073	-12.057	0	0	0	0	0
29	1.00595	-17.136	2.4	0.9	0	0	0
30	0.994506	-18.014	10.6	1.9	0	0	0

APPENDIX C: RESULT OF 3-BUS SYSTEM USING FAULT ANALYSIS

```
-------------Three-phase balanced fault analysis--------------
Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 1.000000e-001
Balanced three-phase fault at bus No. 3
Total fault current =   3.1250 per unit

Bus Voltages during fault in per unit

     Bus     Voltage       Angle
     No.     Magnitude     degrees
      1       0.5938       0.0000
      2       0.6250       0.0000
      3       0.3125       0.0000

Line currents for fault at bus No.  3

     From      To     Current     Angle
     Bus       Bus    Magnitude   degrees
      G         1      1.6250    -90.0000
      1         3      1.8750    -90.0000
      G         2      1.5000    -90.0000
      2         1      0.2500    -90.0000
      2         3      1.2500    -90.0000
      3         F      3.1250    -90.0000

Faultpro v1.0 Balance Three Phase Fault
Universiti Malaysia Pahang
All Rights Reserved
-------------------------Analysis End-------------------------
```

```
-------------Single Line to Ground Fault Analysis------------

Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 1.000000e-001
Single line to-ground fault at bus No. 3
Total fault current =    2.7523 per unit

Bus Voltages during the fault in per unit

     Bus    -------Voltage Magnitude-------
     No.    Phase a     Phase b     Phase c
     1      0.6330      1.0046      1.0046
     2      0.7202      0.9757      0.9757
     3      0.2752      1.0647      1.0647

Line currents for fault at bus No.  3

     From      To      -----Line Current Magnitude----
     Bus       Bus     Phase a     Phase b     Phase c
      1         3      1.6514      0.0000      0.0000
      2         1      0.3761      0.1560      0.1560
      2         3      1.1009      0.0000      0.0000
      3         F      2.7523      0.0000      0.0000

Faultpro v1.0 Single Line to Ground Fault
Universiti Malaysia Pahang
All Rights Reserved
-------------------------Analysis End-------------------------
```

```
------------Double line-to-ground fault analysis-------------

Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 1.000000e-001
Double line-to-ground fault at bus No. 3
Total fault current =    1.9737 per unit

Bus Voltages during the fault in per unit

     Bus    -------Voltage Magnitude-------
     No.    Phase a     Phase b     Phase c
     1      1.0066      0.5088      0.5088
     2      0.9638      0.5740      0.5740
     3      1.0855      0.1974      0.1974

Line currents for fault at bus No.  3

     From      To      -----Line Current Magnitude----
     Bus       Bus     Phase a     Phase b     Phase c
      1         3      0.0000      2.4350      2.4350
      2         1      0.1118      0.3682      0.3682
      2         3      0.0000      1.6233      1.6233
      3         F      0.0000      4.0583      4.0583

Faultpro v1.0 Double line-to-ground fault
Universiti Malaysia Pahang
All Rights Reserved
-------------------------Analysis End-------------------------
```

```
-----------------Line-to-line fault analysis-----------------

Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 1.000000e-001
Line-to-line fault at bus No. 3
Total fault current =    3.2075 per unit

Bus Voltages during the fault in per unit

     Bus    -------Voltage Magnitude-------
     No.    Phase a     Phase b     Phase c
     1      1.0000      0.6720      0.6720
     2      1.0000      0.6939      0.6939
     3      1.0000      0.5251      0.5251

Line currents for fault at bus No.  3

     From      To      -----Line Current Magnitude----
     Bus       Bus     Phase a     Phase b     Phase c
      1         3      0.0000      1.9245      1.9245
      2         1      0.0000      0.2566      0.2566
      2         3      0.0000      1.2830      1.2830
      3         F      0.0000      3.2075      3.2075

Faultpro v1.0 Line-to-line fault
Universiti Malaysia Pahang
All Rights Reserved
-------------------------Analysis End-------------------------
```

APPENDIX D: RESULT ON 11-BUS NETWORK

```
-------------Three-phase balanced fault analysis--------------
Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 0
Balanced three-phase fault at bus No. 8
Total fault current =   3.1392 per unit

Bus Voltages during fault in per unit
```

Bus No.	Voltage Magnitude	Angle degrees
1	0.8137	0.0000
2	0.7578	0.0000
3	0.6923	0.0000
4	0.7472	0.0000
5	0.7201	0.0000
6	0.5500	0.0000
7	0.6120	0.0000
8	0.0000	0.0000
9	0.3039	0.0000
10	0.8351	0.0000
11	0.7229	0.0000

```
Line currents for fault at bus No.  8
```

From Bus	To Bus	Current Magnitude	Angle degrees
G	1	0.9315	-90.0000
1	2	0.9315	-90.0000
2	3	0.2184	-90.0000
2	5	0.2514	-90.0000
2	6	0.4617	-90.0000
3	6	0.3556	-90.0000
4	3	0.1373	-90.0000
4	6	0.3286	-90.0000
4	9	0.6332	-90.0000
5	7	0.2514	-90.0000
6	8	1.1459	-90.0000
7	8	1.3600	-90.0000
8	F	3.1392	-90.0000
9	8	0.6332	-90.0000
G	10	1.0991	-90.0000
10	4	1.0991	-90.0000
G	11	1.1086	-90.0000
11	7	1.1086	-90.0000

-------------Single Line to Ground Fault Analysis------------

Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 0
Single line to-ground fault at bus No. 8
Total fault current = 2.7230 per unit

Bus Voltages during the fault in per unit

Bus	-------Voltage Magnitude-------		
No.	Phase a	Phase b	Phase c
1	0.8923	0.9742	0.9742
2	0.8413	0.9753	0.9753
3	0.7521	0.9907	0.9907
4	0.7858	0.9975	0.9975
5	0.7926	0.9828	0.9828
6	0.5993	1.0052	1.0052
7	0.6583	1.0026	1.0026
8	0.0000	1.0724	1.0724
9	0.3349	1.0320	1.0320
10	0.8568	1.0001	1.0001
11	0.8397	0.9624	0.9624

Line currents for fault at bus No. 8

From	To	-----Line Current Magnitude----		
Bus	Bus	Phase a	Phase b	Phase c
1	2	0.5387	0.2693	0.2693
2	3	0.2119	0.0225	0.0225
2	6	0.4024	0.0019	0.0019
3	6	0.2938	0.0147	0.0147
4	3	0.0819	0.0372	0.0372
4	6	0.2625	0.0225	0.0225
4	9	0.5430	0.0062	0.0062
5	2	0.2351	0.0170	0.0170
6	8	0.9587	0.0353	0.0353
7	5	0.2351	0.0170	0.0170
7	8	1.2213	0.0416	0.0416
8	F	2.7230	0.0000	0.0000
9	8	0.5430	0.0062	0.0062
10	4	0.8875	0.0659	0.0659
11	7	0.6411	0.3205	0.3205

------------Double line-to-ground fault analysis-------------

Fault Impedance (Real) = 0

Fault Impedance (Imaginary)*j = 0

Double line-to-ground fault at bus No. 8

Total fault current = 2.4043 per unit

Bus Voltages during the fault in per unit

Bus	-------Voltage Magnitude-------		
No.	Phase a	Phase b	Phase c
1	0.9524	0.8505	0.8505
2	0.9546	0.7972	0.7972
3	0.9833	0.7201	0.7201
4	0.9955	0.7648	0.7648
5	0.9688	0.7541	0.7541
6	1.0092	0.5730	0.5730
7	1.0045	0.6334	0.6334
8	1.1170	0.0000	0.0000
9	1.0541	0.3185	0.3185
10	1.0002	0.8449	0.8449
11	0.9292	0.7796	0.7796

Line currents for fault at bus No. 8

From	To	-----Line Current Magnitude----		
Bus	Bus	Phase a	Phase b	Phase c
1	2	0.2378	0.8154	0.8154
2	3	0.0198	0.2156	0.2156
2	5	0.0150	0.2445	0.2445
2	6	0.0017	0.4379	0.4379
3	6	0.0130	0.3317	0.3317
4	3	0.0328	0.1205	0.1205
4	6	0.0199	0.3037	0.3037
4	9	0.0055	0.5974	0.5974
5	7	0.0150	0.2445	0.2445
6	8	0.0312	1.0729	1.0729
7	8	0.0367	1.3031	1.3031
8	F	0.0000	2.9725	2.9725
9	8	0.0055	0.5974	0.5974
10	4	0.0582	1.0186	1.0186
11	7	0.2830	0.9704	0.9704

-----------------Line-to-line fault analysis-----------------

Fault Impedance (Real) = 0
Fault Impedance (Imaginary)*j = 0
Line-to-line fault at bus No. 8
Total fault current = 2.7186 per unit

Bus Voltages during the fault in per unit

Bus	-------Voltage Magnitude-------		
No.	Phase a	Phase b	Phase c
1	1.0000	0.8641	0.8641
2	1.0000	0.8251	0.8251
3	1.0000	0.7807	0.7807
4	1.0000	0.8178	0.8178
5	1.0000	0.7993	0.7993
6	1.0000	0.6906	0.6906
7	1.0000	0.7286	0.7286
8	1.0000	0.5000	0.5000
9	1.0000	0.5651	0.5651
10	1.0000	0.8793	0.8793
11	1.0000	0.8012	0.8012

Line currents for fault at bus No. 8

From	To	-----Line Current Magnitude----		
Bus	Bus	Phase a	Phase b	Phase c
1	2	0.0000	0.8067	0.8067
2	3	0.0000	0.1891	0.1891
2	5	0.0000	0.2177	0.2177
2	6	0.0000	0.3998	0.3998
3	6	0.0000	0.3080	0.3080
4	3	0.0000	0.1189	0.1189
4	6	0.0000	0.2846	0.2846
4	9	0.0000	0.5484	0.5484
5	7	0.0000	0.2177	0.2177
6	8	0.0000	0.9924	0.9924
7	8	0.0000	1.1778	1.1778
8	F	0.0000	2.7186	2.7186
9	8	0.0000	0.5484	0.5484
10	4	0.0000	0.9518	0.9518
11	7	0.0000	0.9601	0.9601

APPENDIX E: SOLUTION OF OPTIMAL DISPATCH WITH POWER FLOW

Maximum Power Mismatch = 1.90285-008

No of Iteration = 2

Bus No	Voltage Magnitude	Angle Degree	Load		Generation		Injected
			MW	MVar	MW	MVar	MVar
1	1.060	0.0000	0.0	0.0	23.6488	25.7266	0.000
2	1.045	-0.2819	20	10	69.5182	30.7666	0.000
3	1.03	-0.4946	20	15	58.9899	14.0522	0.000
4	1.01863	-1.2075	50	30	0	0	0
5	0.99014	-2.7290	60	40	0	0	0

Total Generation (MW) = 152.157

Total Generation (MVar) = 70.5454

Total Load (MW) = 150

Total Load (MVar) = 95

Total System Loss (MW) = 2.15434

Incremental cost of delivered power (system lambda) = 7.75905 $/MWh

Optimal Dispatch of Generation

23.5581

69.5593

59.0368

Total Generation Cost = 1596.96 $/h

APPENDIX F: M-File of Main Page of the Project

```
function varargout = PSA(varargin)
% PSA M-file for PSA.fig
%      PSA, by itself, creates a new PSA or raises the existing
%      singleton*.
%
%      H = PSA returns the handle to a new PSA or the handle to
%      the existing singleton*.
%
%      PSA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in PSA.M with the given input arguments.
%
%      PSA('Property','Value',...) creates a new PSA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before PSA_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to PSA_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help PSA

% Last Modified by GUIDE v2.5 31-May-2008 22:20:08

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @PSA_OpeningFcn, ...
                   'gui_OutputFcn',  @PSA_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before PSA is made visible.
function PSA_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PSA (see VARARGIN)

% Choose default command line output for PSA
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PSA wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = PSA_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(aboutdefinition)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(main_menu)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure (faultpro)
```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure (main_page)

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(optimal)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

delete (handles.figure1)
```

APPENDIX G: M-File of Load flow Analysis (Example of 3-bus network)

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(main_menu)
delete(example3result1)
delete(example3a)
delete(example3)
delete(example)
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(example)
delete(example3result1)
delete(example3a)
delete(example3)

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
busdata=[1   1   1.05  0    0    0    0    0    0    0    0
         2   2   1.03  0    50   20   20   0    0    0    0
         3   0   1.0   0    60   25   0    0    0    0    0];


 linedata=[1   2   0.08   0.24   0     1
           1   3   0.02   0.06   0     1
           2   3   0.06   0.18   0     1];

 basemva = 100;
 accuracy = 0.001;
 accel = 1.8;
 maxiter = 100;
```

```
Lfybus                              % form the bus admittance matrix
Lfnewton                  % Load flow solution by newton-raphson method
Busout                   % Prints the power flow solution on the screen
Lineflow1

set(handles.edit6,'String',iter)
set(handles.edit1,'String',maxerror)
set(handles.edit2,'String',Pgt)
set(handles.edit3,'String',Qgt)
set(handles.edit4,'String',Pdt)
set(handles.edit5,'String',Qdt)
set(handles.edit7,'String',real(SLT))
set(handles.edit8,'String',imag(SLT))
figure(PFAresultex3)
delete (example3a)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
busdata=[1  1  1.05  0   0   0   0   0   0   0   0
         2  2  1.03  0   50  20  20  0   0   0   0
         3  0  1.0   0   60  25  0   0   0   0   0];

linedata=[1  2  0.08  0.24  0    1
          1  3  0.02  0.06  0    1
          2  3  0.06  0.18  0    1];

basemva = 100;
accuracy = 0.001;
accel = 1.8;
maxiter = 100;
```

```
Lfybus                           % form the bus admittance matrix
lfgauss                    % Load flow solution by Gauss Siedel method
Busout                   % Prints the power flow solution on the screen
Lineflow1

set(handles.edit6,'String',iter)
set(handles.edit1,'String',maxerror)
set(handles.edit2,'String',Pgt)
set(handles.edit3,'String',Qgt)
set(handles.edit4,'String',Pdt)
set(handles.edit5,'String',Qdt)
set(handles.edit7,'String',real(SLT))
set(handles.edit8,'String',imag(SLT))
figure(PFAresult1ex3)
delete (example3a)

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
busdata=[1  1  1.05  0   0   0   0   0   0   0   0
         2  2  1.03  0   50  20  20  0   0   0   0
         3  0  1.0   0   60  25  0   0   0   0   0];

linedata=[1  2  0.08  0.24  0    1
          1  3  0.02  0.06  0    1
          2  3  0.06  0.18  0    1];

basemva = 100;
accuracy = 0.001;
accel = 1.8;
maxiter = 100;
```

```
Lfybus                              % form the bus admittance matrix
Decouple                 % Load flow solution by fast decoupled method
Busout                  % Prints the power flow solution on the screen
Lineflow1

set(handles.edit6,'String',iter)
set(handles.edit1,'String',maxerror)
set(handles.edit2,'String',Pgt)
set(handles.edit3,'String',Qgt)
set(handles.edit4,'String',Pdt)
set(handles.edit5,'String',Qdt)
set(handles.edit7,'String',real(SLT))
set(handles.edit8,'String',imag(SLT))
figure(PFAresult2ex3)
delete (example3a)


% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
commandwindow

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
commandwindow

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
commandwindow
```

APPENDIX H: M-File of Optimal Dispatch (Sudden Change in Power)

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Get user input from GUI
P0= str2double(get(handles.P0,'String'));
E = str2double(get(handles.E,'String'));
V = str2double(get(handles.V,'String'));
X = str2double(get(handles.X,'String'));

Pemax= E*V/X;
if P0  >= Pemax

return, end
d0=asin(P0/Pemax);
delta = 0:.01:pi;
Pe = Pemax*sin(delta);
dmax=pi;
Ddmax=1;
while abs(Ddmax) > 0.00001
Df = cos(d0) - (sin(dmax)*(dmax-d0)+cos(dmax));
J=cos(dmax)*(dmax-d0);
Ddmax=Df/J;
dmax=dmax+Ddmax;
end

dc=pi-dmax;
Pm2=Pemax*sin(dc);
Pmx =[0  pi-d0]*180/pi; Pmy=[P0 P0];
Pm2x=[0  dmax]*180/pi; Pm2y=[Pm2 Pm2];
x0=[d0 d0]*180/pi; y0=[0 Pm2]; xc=[dc dc]*180/pi; yc=[0 Pemax*sin(dc)];
xm=[dmax dmax]*180/pi; ym=[0 Pemax*sin(dmax)];
d0=d0*180/pi; dmax=dmax*180/pi; dc=dc*180/pi;
x=(d0:.1:dc);

 y=Pemax*sin(x*pi/180);
 %y1=Pe2max*sin(d0*pi/180);
 %y2=Pe2max*sin(dc*pi/180);
```

```
x=[d0 x dc];
y=[Pm2 y Pm2];
xx=dc:.1:dmax;
h=Pemax*sin(xx*pi/180);
xx=[dc xx  dmax];
hh=[Pm2 h  Pm2];
delta=delta*180/pi;
%calculate the parameters
fprintf('\nInitial power                       =%7.3f p.u.\n', P0)
fprintf('Initial power angle                 =%7.3f degrees \n', d0)
fprintf('Sudden additional power             =%7.3f p.u.\n', Pm2-P0)
fprintf('Total power for critical stability =%7.3f p.u.\n', Pm2)
fprintf('Maximum angle swing                 =%7.3f degrees \n', dmax)
fprintf('New operating angle                 =%7.3f degrees \n\n\n', dc)

%Ploting the graph
h = figure; figure(h);
fill(x,y,'m')
hold;
fill(xx,hh,'c')
plot(delta, Pe,'-', Pmx, Pmy,'g', Pm2x,Pm2y,'g', x0,y0,'c', xc,yc, xm,ym,'r'), grid
title('Equal-area criterion applied to the sudden change in power')
xlabel('Power angle, degree'), ylabel(' Power, per unit')
axis([0 180  0 1.1*Pemax])
hold off;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
figure(Type_of_Equal_Area_Criterion_Application)
delete(EAC_Sudden_Change_In_Power)

% --------------------------------------------------------------------
function Help_menu_Callback(hObject, eventdata, handles)
% hObject    handle to Help_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
!notepad Application_of_Equal_Area_Criterion_due_to_sudden_change_in_power.txt
```

```
% --------------------------------------------------------------------
function Exit_menu_Callback(hObject, eventdata, handles)
% hObject    handle to Exit_menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get the current position of the GUI from the handles structure
% to pass to the modal dialog.
pos_size = get(handles.figure1,'Position');
% Call modaldlg with the argument 'Position'.
user_response = exit('Title','Confirm Close');
switch user_response
case {'No'}
% take no action
case 'Yes'
% Prepare to close GUI application window
% .
% .
% .
quit
end
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
!notepad Application_of_Equal_Area_Criterion_due_to_sudden_change_in_power.txt
```